

A Market for Scheduling, with Applications to Cloud Computing

Nikhil Devanur* Jugal Garg† Ruta Mehta‡ Vijay V. Vazirani§ Sadra Yazdanbod¶

Abstract

We present a market for allocating and scheduling resources to agents who have specified budgets and need to complete specific tasks. Two important aspects required in this market are: (1) agents need specific amounts of each resource to complete their tasks, and (2) agents would like to complete their tasks as soon as possible. In incorporating these aspects, we arrive at a model that deviates substantially from market models studied so far in economics and theoretical computer science. Indeed, all known techniques developed to compute equilibria in markets in the last decade and half seem not to apply here.

We give a polynomial time algorithm for computing an equilibrium using a new technique that is somewhat reminiscent of the *ironing* procedure used in the characterization of optimal auctions by Myerson. This is in spite of the fact that the set of equilibrium prices could be non-convex; in fact it could have “holes”. Our market model is motivated by the cloud computing marketplace. Even though this market is already huge and is projected to grow at a massive rate, it is currently run in an ad hoc manner.

*Microsoft Research. Part of the work was done when author was visiting Simons Institute, UC Berkeley.
nikdev@microsoft.com

†Max-Planck-Institut für Informatik, Saarbrücken, Germany. jugal.garg@gmail.com

‡Simons Institute for the Theory of Computing, UC Berkeley. mehta.ruta@gmail.com

§College of Computing, Georgia Institute of Technology, Atlanta. vazirani@cc.gatech.edu

¶College of Computing, Georgia Institute of Technology, Atlanta. syazdanb@cc.gatech.edu

1 Introduction

The Internet has transformed not only the economy but also the central object of study of economists, the market, by creating innovative and immensely important marketplaces. Indeed, quoting Papadimitriou [35]:

The Internet ... transformed, informed and accelerated markets, while creating new, theretofore unimaginable kinds of market—in addition to being itself, in important ways, a market.

The most important such market so far was the multi-billion dollar adwords market, and researchers from algorithms and AGT have contributed handsomely to its efficient operation [33, 16, 18, 39, 17, 1]. A quickly emerging market is the cloud computing market. Since most projections predict that this market will dwarf even the adwords market, it is quintessential to understand its idiosyncrasies and design algorithms and mechanisms for its efficient operation.

The *Elastic Cloud Computing (EC2)* market of Amazon is the biggest provider of cloud computing resources today. The EC2 market rents out a number of different types of resources – virtual machines (VM) with different kinds of capabilities, e.g., compute optimized, storage optimized, memory optimized and general purpose. While this market is still in its infancy, as is the science behind it, future growth in the size and complexity of this market calls for mechanisms that are steeped in sound economic theory and the theory of algorithms.

The power of the pricing mechanism is well explored and understood in economics: It allocates resources efficiently since prices send strong signals about what is wanted and what is not, and it prevents artificial scarcity of goods while at the same time ensuring that goods that are truly scarce are conserved. Hence it is beneficial to both consumers and producers. For this reason, we propose an equilibrium-based mechanism for a spot market for cloud computing resources.¹ In addition, we give a polynomial time algorithm to find an equilibrium.

Two important aspects required in this model are: (1) agents desire *duration guarantees*: they must be allocated each resource for a specified amount of time, and (2) agents would like to complete their tasks as soon as possible. In incorporating these aspects, we arrive at a model that deviates substantially from market models studied so far in economics and theoretical computer science. The latter models define preferences of agents via a utility function: an agent prefers that bundle of goods which maximizes a given (usually concave) utility function, subject to a budget constraint; strict constraints on goods are never allowed. In contrast, our model allows agents to state their exact requirement for each good, and satisfies these covering requirements fully. In addition, our model incorporates a temporal aspect, i.e., there is a notion of time in the model and prices of resources are a function of time. These features make our market model suitable for resource allocation and scheduling, which are central to cloud computing.

These differences from standard models manifest themselves in a fundamental way: the set of equilibrium prices could form a non-convex, yet connected region. This immediately rules out a convex programming based approach, and in fact indicates that the problem is computationally hard. Hence it was quite surprising that the problem did admit a polynomial time algorithm.

1.1 Our Model and Results

Let A be a set of n agents, indexed by i , and G be a set of m different types of goods, indexed by j . Time is divided into *slots*, each of one time unit. Assume that there are s slots and that they are indexed by t . Each

¹Indeed, Section 1.3 gives examples of resource allocation problems where a market equilibrium based mechanism has been used even when there is no market per se and there are no monetary transfers.

agent i has a requirement of r_{ij} (≥ 0) units of good j . For each slot t we are specified a set G^t , the set of goods that are available in this slot. For each good $j \in G$, a positive weight w_j is specified, representing the relative importance of this good. Let f_{ijt} be the allocation of good j to agent i in slot t . An *allocation is feasible* if it assigns the required amount of each good to each agent, i.e., for each $i \in A$ and $j \in G$, $\sum_t f_{ijt} = r_{ij}$, and it assigns at most one unit of good j in slot t if $j \in G^t$ and does not assign any of good j in slot t otherwise. Each agent i wants to minimize her own *weighted flow time*, which is defined to be

$$\sum_j \sum_t w_j t f_{ijt}.$$

Suppose that we allocate the goods to agents using a market mechanism: agent i has a total budget of m_i , and let p_{jt} denote the price per unit amount of good j in slot t . Then, the *amount of money charged* to agent i for this allocation is $\sum_{j,t} f_{ijt} \cdot p_{jt}$. We will say that agent i gets an *optimal allocation* if relative to prices p , agent i minimizes her weighted flow time, subject to the budget constraint that the allocation does not cost more than her budget. Finally, we will say that a feasible allocation and prices (f, p) are an *equilibrium* if the following two conditions hold:

1. Each agent gets an optimal allocation relative to prices p .
2. If in some slot t a good $j \in G^t$ is not fully allocated, i.e., to the extent of one time unit, then $p_{jt} = 0$.

We also define a model where time is continuous, and prices can change at any time. The flow time of an agent is also calculated continuously, as an integral.

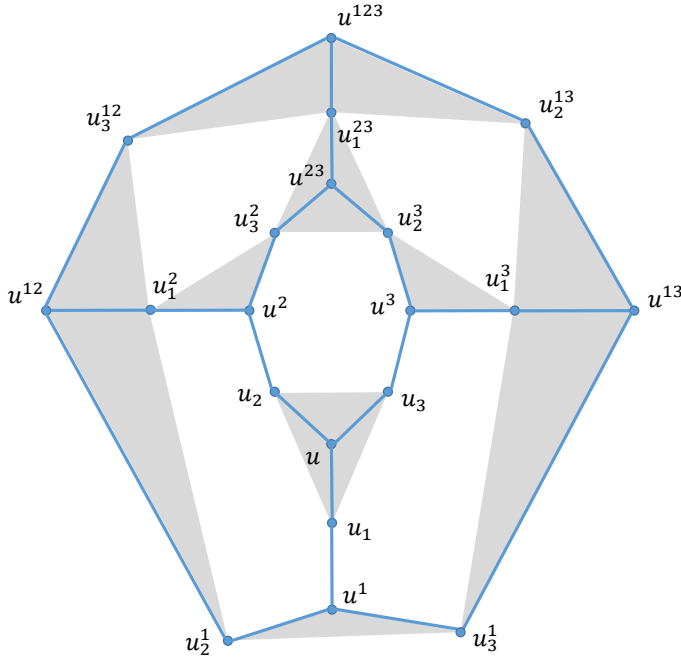
We provide a polynomial time algorithm to compute an equilibrium in this market. As with the classical Arrow-Debreu and Fisher market models, we show that our market model satisfies the First Welfare Theorem, stating that equilibrium allocations are Pareto optimal. Additionally, we show that our market based mechanism is *incentive compatible*, i.e., it is in the best interest of the agents to truthfully report m_i and r_{ij} s.

Our proposal for the cloud computing spot market application is to run our market periodically, each time allocating resources to all agents who are currently seeking them. It may happen that some of the resources in the early slots of the current period are already allocated in the previous run of the market. Observe that our model is general enough — it allows G^t to be a proper subset of G — hence it will only allocate the remaining goods in these slots.

Non convex solution sets: A natural question, given the strong connection between equilibria in Fisher markets and convex programs, is whether there exists a convex program that captures the equilibrium allocation/prices in this market. We show via examples that this is unlikely to be the case. In fact, the set of equilibrium prices could have “holes” in between, i.e., this set could have genus > 0 . We show an example of a set with a high genus in Figure 1 on page 3. Given this, it is surprising that we are able to find a combinatorial polynomial time algorithm for this problem. As far as we are aware, there is no other natural example where such a phenomenon happens.

1.2 Techniques

The non-convexity of the set of equilibrium prices is a good indication of the difficulty involved here. For instance, this immediately rules out any convex programming based solution. All other known techniques developed to compute equilibria in markets in the last decade and half don’t seem to apply here. The distinguishing feature of our market is the fact that the buyer’s optimization problem contains a covering as well as a packing constraint. Due to this, the preferences of the buyers look very different from any quasi-concave utility based preference. (See Figure 2 on page 4 for an example.)



u :	$(56, 44, 33, 23, 16, 10, 5, 3, 2)$
u_1 :	$(56, 44^{1/3}, 32^{2/3}, 23, 16, 10, 5, 3, 2)$
u_2 :	$(56, 44, 33, 23, 16^{1/3}, 9^{2/3}, 5, 3, 2)$
u_3 :	$(56, 44, 33, 23, 16, 10, 5, 3^{1/3}, 1^{2/3})$
u^1 :	$(56^{2/3}, 44^{1/3}, 32, 23, 16, 10, 5, 3, 2)$
u^2 :	$(56, 44, 33, 23^{2/3}, 16^{1/3}, 9, 5, 3, 2)$
u^3 :	$(56, 44, 33, 23, 16, 10, 5^{2/3}, 3^{1/3}, 1)$
u_1^1 :	$(56^{2/3}, 44^{1/3}, 32, 23, 16^{1/3}, 9^{2/3}, 5, 3, 2)$
u_1^3 :	$(56^{2/3}, 44^{1/3}, 32, 23, 16, 10, 5, 3^{1/3}, 1^{2/3})$
u_2^1 :	$(56, 44^{1/3}, 32^{2/3}, 23^{2/3}, 16^{1/3}, 9, 5, 3, 2)$
u_2^3 :	$(56, 44, 33, 23^{2/3}, 16^{1/3}, 9, 5, 3^{1/3}, 1^{2/3})$
u_3^1 :	$(56, 44^{1/3}, 32^{2/3}, 23, 16, 10, 5^{2/3}, 3^{1/3}, 1)$
u_2^3 :	$(56, 44, 33, 23, 16^{1/3}, 9^{2/3}, 5^{2/3}, 3^{1/3}, 1)$
u^{12} :	$(56^{2/3}, 44^{1/3}, 32, 23^{2/3}, 16^{1/3}, 9, 5, 3, 2)$
u^{23} :	$(56, 44, 33, 23^{2/3}, 16^{1/3}, 9, 5^{2/3}, 3^{1/3}, 1)$
u^{13} :	$(56^{2/3}, 44^{1/3}, 32, 23, 16, 10, 5^{2/3}, 3^{1/3}, 1)$
u_3^{12} :	$(56^{2/3}, 44^{1/3}, 32, 23^{2/3}, 16^{1/3}, 9, 5, 3^{1/3}, 1^{2/3})$
u_1^{23} :	$(56, 44^{1/3}, 32^{2/3}, 23^{2/3}, 16^{1/3}, 9, 5^{2/3}, 3^{1/3}, 1)$
u_2^{13} :	$(56^{2/3}, 44^{1/3}, 32, 23, 16^{1/3}, 9^{2/3}, 5^{2/3}, 3^{1/3}, 1)$
u^{123} :	$(56^{2/3}, 44^{1/3}, 32, 23^{2/3}, 16^{1/3}, 9, 5^{2/3}, 3^{1/3}, 1)$

Figure 1: An example where the set of equilibrium prices has genus 5. There is 1 good and there are 9 buyers, each with a requirement of 1. Their m_i s are 56, 44, 33, 23, 16, 10, 5, 3 and 2. All prices on the shaded region and on the boundary are equilibria. However, points on the non-shaded region are not equilibria, say u^2 and u^3 in the figure, then none of the points in their convex combination is an equilibrium.

For a special case of our problem, with one good and when the requirements are all 1, we show that equilibrium conditions are equivalent to a set of conditions that are reminiscent of the *ironing* procedure used in the characterization of optimal auctions by Myerson [34]. It is in fact “one higher derivative” analog of Myerson’s ironing, which given a possibly non-monotone function, asks for an ironed function that is *monotone non-increasing*, and is such that the area under the curve (starting at 0) of the ironed function is always higher than that for the given function. Further, the ironed function given by this procedure is the *minimal* among all such functions. This means that wherever the area under the curve differs for the two functions, the ironed function is *constant*. (See Figure 3 on page 5.)

In the special case of our model with one good and unit requirements, the equilibrium price of the good as a function of time is obtained as an ironed analog of the *money function*: the function $i \mapsto m_i$, where we assume the m_i s are sorted in the decreasing order. This money function is monotone non-increasing by definition but it need not be a convex function. The price as a function of time must be a *monotone non-increasing and convex* function. The area under the curve of the price function must always be higher than that of the money function; further, wherever the two areas are different, the price function must be *linear*. One can see that the conditions are the same as that of Myerson’s ironing, except each condition is replaced by a higher derivative analog. Unlike Myerson’s, the solution to our problem is no longer unique!

The story for multiple goods is a lot more complicated. Here there are multiple price functions, one for each good, but only a single money function. We observe that the right thing to do here is to search in the space of price slopes, or equivalently exchange rates between cost and delay for each buyer. It turns out that the optimal way for a buyer to spread his money among different goods is to combine delay and price linearly using a coefficient λ_i for delay (and 1 for price), and buy the cheapest r_{ij} slots of good j according to this. Buyers who have a higher λ_i get scheduled earlier. In each iteration of the algorithm, we identify

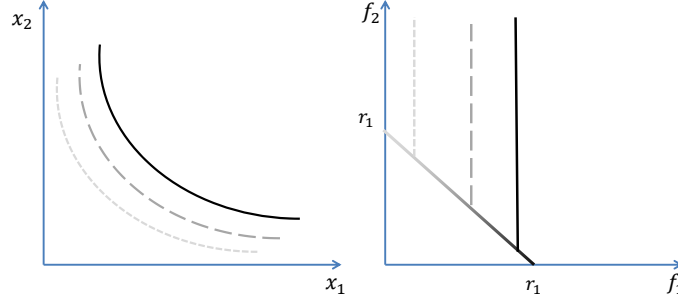


Figure 2: On the left is a typical example of indifference curves with a quasi concave utility function over two variables. On the right are indifference curves/regions for the case of 1 good and 2 slots. Everything below the 45 degree line is one region, and is the least preferred, since the buyer does not get his required amount of resources here. Along the 45 degree line, there is a total order, with the point on the x axis being the most preferred and the point on the y axis being the least preferred. The indifference curves are now axis parallel lines, since getting more than his requirement doesn't matter for the buyer. The entire region to the right of the point $(r_1, 0)$ is the most preferred region.

the set of buyers who are going to have the smallest λ_i s, and get the latest of the slots. This requires doing a binary search where to determine which direction the binary search moves, a submodular minimization problem must be solved. We show that we can set the prices of these slots so that it doesn't conflict with our future choices, and recurse with the remaining buyers.

1.3 Other Applications and Related Work

There is a long history of market based mechanisms: the New York Stock Exchange uses such a mechanism to determine the opening prices, and copper and gold prices in London are fixed using a similar procedure [37]. Hurwicz [24] showed that strategic behavior by agents participating in such a mechanism can lead to inefficiencies. Babaioff et al. [3] show price of anarchy bounds on such mechanisms.

Fair allocation Market equilibrium outcomes have been used to allocate resources by a central planner seeking a fair allocation, even when there is no actual market/monetary transfers. Equilibrium conditions are often considered inherently fair, due to the properties mentioned earlier. The following are some examples.

- The proportional fair allocation is widely used in the design of computer networks. It is a well known fact that this is equivalent to the equilibrium allocation in a Fisher market [31].
- Budish [5] proposes “competitive outcome from equal incomes” (CEEI) as a way to allocate courses to students: the allocation is an equilibrium in a market for courses in which the students participate

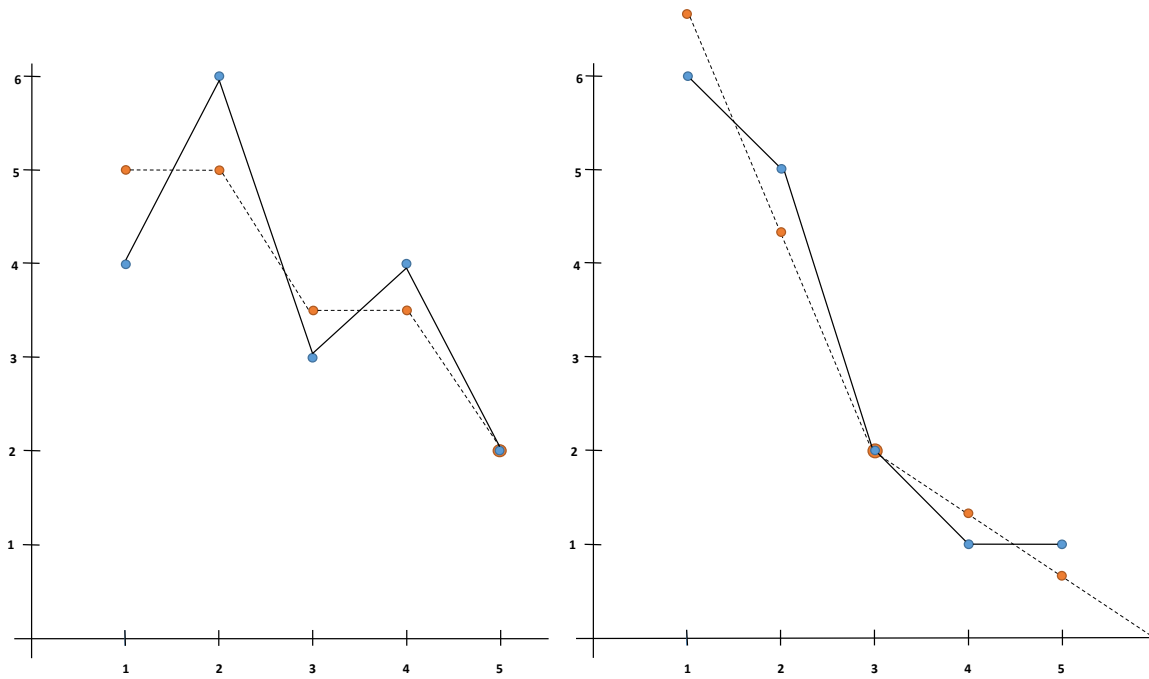


Figure 3: On the left is an example of Myerson’s ironing. The solid curve with the blue dots is the given curve, which is non-monotone. The dashed curve with the orange dots is the ironed curve, which is monotone. On the right is an example of our problem. The solid curve is the money function, which is monotone but not convex. The dashed one is the price function, which is convex. Both dashed curves are such that their “area under the curve” is higher than that for the solid curves, and satisfy a minimality condition among all such curves.

with equal budgets (with random perturbations to break ties). This scheme has been successfully used at the Wharton business school [6].

- Cole et al. [11] show that a suitable modification of the Fisher market equilibrium allocation can be used as a solution to a problem of fair resource allocation, without money. The mechanism is truthful, and satisfies an approximate per-agent welfare guarantee. The motivation for [11] is also allocation of cloud computing resources, among agents internal to a company running its own cloud services, but it does not capture the temporal aspect.

Fair allocation of resources in a scheduling setting has received a lot of attention lately [23, 29, 36, 41]. However, most of these focus on *instantaneous* fairness, which ignores the important temporal aspect of jobs. Our market model suggests a natural fair allocation mechanism: assign m_i s based on agents’ relative importance², and use the resulting equilibrium allocation.

Equilibrium outcomes have also been used in the design of online algorithms in a scheduling context by Im et al. [25, 26]. They give the first constant competitive online algorithms in a multi-resource scheduling framework for the objectives of completion time and flow time. It is possible that equilibria in our model will find similar applications, given that it already incorporates the scheduling aspect.

There has been a lot of activity in the theoretical computer science literature on algorithms for and hardness of computing market equilibria [12, 15, 13, 19, 14, 22, 27, 28, 20, 40, 10, 8, 7]. As mentioned

²These can all be equal, with some small random perturbations to break ties, just like in the CEEI allocation.

before, our market model seems to be quite different from these markets in terms of computability.

2 Preliminaries

Consider the market \mathcal{M} described in Section 1.1, with set A of agents and set G of resources/goods. Resources are available in unit amount over a period of time. Time is divided into slots, each with unit time, and G^t is the set of resources available in slot t . Each agent $i \in A$ has budget m_i and needs resource j for total of r_{ij} amount. Thus total requirement for good j is $R_j = \sum_i r_{ij}$. We assume that there are enough slots to fulfill everyone's requirement.

Relative importance of good j is specified by w_j , in the sense that if f_{ijt} denotes the amount of good j agent i gets in slot t , then she wishes to minimize her weighted flow time $\sum_{jt} w_j t f_{ijt}$. Therefore, at prices p_{jt} for good j in slot t , agent i 's optimal demand bundle may be computed by the following linear program (LP).

$$\begin{aligned} \min : & \sum_{jt} w_j t f_{ijt} \\ \text{s.t. : } & \sum_t f_{ijt} \geq r_{ij}, & \forall j \in G \\ & \sum_{t, j \in G^t} p_{jt} f_{ijt} \leq m_i \\ & 0 \leq f_{ijt} \leq 1, & \forall t, \forall j \in G^t \end{aligned} \tag{1}$$

Prices \mathbf{p} are said to be at equilibrium, if when each agent is given its optimal bundle, market clears. In other words total demand of every good in each slot is at most one, and under-demanded slots are priced zero. Formally,

$$\begin{aligned} \forall t, \forall j \in G^t : & \sum_{i \in A} f_{ijt} \leq 1 \\ \forall t, \forall j \in G^t : & \sum_{i \in A} f_{ijt} < 1 \Rightarrow p_{jt} = 0 \end{aligned} \tag{2}$$

Our goal is to find such a prices \mathbf{p} and corresponding allocation \mathbf{f} . As noted in Section 1.1 equilibrium prices may not be convex and therefore convex programming approach is ruled out. Even to show rationality of equilibrium prices, it is not clear if any of the previous approaches, based on equilibrium configuration [40] or LCP [20], are applicable. This is because, both budget and covering constraints being present in the optimal bundle LP. Therefore it becomes necessary to understand characterization of equilibria for this model, done in the next section.

From now on we assume that if $r_{ij} \neq 0$ then it is at least one.³ For notational simplicity we will also assume that $G^t = G$, $\forall t$ for now, and later in Section 5 we will show how to extend all the results without this assumption.

3 Equilibrium Characterization

A given market may have many equilibrium prices. In this section we derive sufficient conditions for \mathbf{p} and \mathbf{f} to be an equilibrium. We note that these conditions are not necessary and therefore may end up characterizing a particular type of equilibrium. We start with writing KKT condition for (1) LP. Let β'_{ij} , λ'_i , and γ'_{ijt} be the dual variables of first, second and $f_{ijt} \leq 1$ constraints of this LP. Apart from non-negativity of these variables, the KKT conditions give us:

³Finally it is a matter of pricing resources over time. Therefore, the case when $r_{ij} < 1$ can be handled by reducing the length of each slot to $\min_{i,j, r_{ij} \neq 0} r_{ij}$

$$\begin{aligned}
\forall t, \forall j \in G : \quad & \beta'_{ij} \leq w_j t + p_{jt} \lambda'_i + \gamma'_{ijt} \\
\forall t, \forall j \in G : \quad & f_{ijt} > 0 \Rightarrow \beta'_{ij} = w_j t + p_{jt} \lambda'_i + \gamma'_{ijt} \\
\forall t, \forall j \in G : \quad & \gamma'_{ijt} > 0 \Rightarrow f_{ijt} = 1 \\
& \sum_{t,j \in G} p_{jt} f_{ijt} < m_i \Rightarrow \lambda'_i = 0
\end{aligned} \tag{3}$$

Next lemma characterizes allocation of an agent with $\lambda'_i = 0$.

Lemma 1 *If for an agent $\lambda'_i = 0$, then for all $j \in G$ we have $f_{ijt} = 1, \forall t < r_{ij}$ and $f_{ij} \lceil r_{ij} \rceil = r_{ij} - \lfloor r_{ij} \rfloor$.*

Proof : It suffices to show $\forall t, \forall j \in G$ such that $f_{ijt} > 0$ we have that $\forall \hat{t} < t, f_{ij\hat{t}} = 1$. Using the KKT conditions (3), we get that

$$\beta'_{ij} = w_j t + \gamma'_{ijt} \leq w_j \hat{t} + \gamma'_{ij\hat{t}} \Rightarrow \hat{t} < t \Rightarrow \gamma'_{ij\hat{t}} > \gamma'_{ijt} \geq 0 \Rightarrow_{(3)} f_{ij\hat{t}} = 1$$

□

Lemma 1 and the last condition of (3) implies that only the first agent allocated to a good can under-spend. Using this fact, in the algorithm we will make sure that all the agent spends all their money, and therefore without loss of generality (wlog) we can assume that $\lambda'_i > 0$. Then dividing the first condition of (3) by λ'_i and renaming $\frac{\beta'_{ij}}{\lambda'_i}$ by β_{ij} , $\frac{1}{\lambda'_i}$ by λ_i , and $\gamma'_{ijt} \lambda'_i$ by γ_{ijt} gives:

$$\begin{aligned}
\forall t, \forall j \in G : \quad & \beta_{ij} \leq (w_j t) \lambda_i + p_{jt} + \gamma_{ijt} \\
\forall t, \forall j \in G : \quad & f_{ijt} > 0 \Rightarrow \beta_{ij} = (w_j t) \lambda_i + p_{jt} + \gamma_{ij} \\
\forall t, \forall j \in G : \quad & \gamma_{ijt} > 0 \Rightarrow f_{ijt} = 1 \\
\forall i \in A : \quad & \lambda_i > 0
\end{aligned} \tag{4}$$

Using (4), next we characterize optimal bundle of any given agent.

Lemma 2 *Given prices p_{jt} and λ_i , for every good, agent i demands slots in increasing order of $(w_j t) \lambda_i + p_{jt}$, i.e., if $(w_j t) \lambda_i + p_{jt} < (w_j t') \lambda_i + p_{jt'}$ and $f_{ijt'} > 0$ then $f_{ijt} = 1$.*

Proof : Using (4) and $f_{ijt'} > 0$ we get that $(w_j t') \lambda_i + p_{jt'} = \beta_{ij} - \gamma_{ijt}$. Therefore,

$$\beta_{ij} - \gamma_{ijt} \leq_{(4)} (w_j t) \lambda_i + p_{jt} < \beta_{ij} - \gamma_{ijt'} \Rightarrow \gamma_{ijt} > \gamma_{ijt'} \geq 0 \Rightarrow_{(4)} f_{ijt} = 1$$

□

Lemma 2 shows a relation between two types of costs that an agent has to bear, namely delay and price paid. And λ_i seem to act like a conversion rate between the two, i.e., one unit of delay is equivalent to λ_i dollars. Using this next we show a strong connection between λ_i s and prices.

Assuming $\gamma_{ijt} = 0$: We next force all the γ_{ijt} s to be 0; this is not necessary by any means but we will see that in the end we will still be able to find solutions satisfying this condition. This forces all the slots which an agent buys to have the same combined cost, namely $(w_j t) \lambda_i + p_{jt}$. With this assumption, the complementary slackness conditions (4) are:

$$\begin{aligned}
& \forall t, \forall j \in G : \beta_{ij} \leq (w_j t) \lambda_i + p_{jt} \\
& \forall t, \forall j \in G : f_{ijt} > 0 \Rightarrow \beta_{ij} = (w_j t) \lambda_i + p_{jt} \\
& \forall i \in A : \lambda_i > 0
\end{aligned} \tag{5}$$

Next two lemmas give us a clean characterization of equilibrium prices. We will have to consider demand of a good from a set of agent, defined next.

Definition 3 For a set $S \subseteq A$ of agents, define $r_j(S) = \sum_{i \in S} r_{ij}$ be their total demand for good j .

Lemma 4 At equilibrium $p_{jt} > p_{j(t+1)}$, $\forall t \leq r_j(A)$; $p_{jt} = 0$, $\forall t > r_j(A)$. Further, $p_{j(t-1)} - p_{jt} \geq p_{jt} - p_{j(t+1)}$, $\forall t > 1$.

Proof : Suppose $\exists t$ such that $p_{jt} \leq p_{j(t+1)}$. Since $p_{j(t+1)} > 0$ we have that $\exists i$ such that $f_{ijt} > 0$. Then $\beta_{ij} \stackrel{(5)}{=} (w_j(t+1)) \lambda_i + p_{j(t+1)} > (w_j t) \lambda_i + p_{jt}$ which is a contradiction.

Suppose $\exists t > r_j(A)$ such that $p_{jt} > 0 \Rightarrow \exists i$ s.t. $f_{ijt} > 0$. Note that $\sum_{i,t} f_{ijt} = r_j(A)$. Therefore, $\exists t' \leq r_j(A)$ such that $f_{ijt'} < 1$. This gives us a contradiction considering Lemma 2.

If $p_{jt} = 0$ then we have $p_{j(t-1)} - p_{jt} \geq 0 = p_{jt} - p_{j(t+1)}$. On the other hand if $p_{jt} > 0$ then $\exists i$, $f_{ijt} > 0 \Rightarrow \stackrel{(5)}{\beta_{ij}} = (w_j t) \lambda_i + p_{jt}$. Using $\beta_{ij} \leq (w_j(t-1)) \lambda_i + p_{j(t-1)}$ and $\beta_{ij} \leq (w_j(t+1)) \lambda_i + p_{j(t+1)}$, it is easy to show $p_{j(t-1)} - p_{jt} \geq \lambda_i \geq p_{jt} - p_{j(t+1)}$. \square

Lemma 5 For good j , if the first and last slot bought by agent i are s and s' , such that $s < s'$, then $p_{jt} - p_{j(t+1)} = w_j \lambda_i$, $\forall s \leq t < s'$; $p_{jt} - p_{j(t+1)} \geq w_j \lambda_i$, $\forall t < s$, and $p_{jt} - p_{j(t+1)} \leq w_j \lambda_i$, $\forall t \geq s'$.

Proof : We have that

$$\begin{aligned}
\beta_{ij} &= (w_j s) \lambda_i + p_{js} \leq (w_j(s-1)) \lambda_i + p_{j(s-1)} \Rightarrow p_{j(s-1)} - p_{js} \geq w_j \lambda_i \\
&\Rightarrow \text{Lemma 4 } \forall t < s, p_{jt} - p_{j(t+1)} \geq w_j \lambda_i
\end{aligned}$$

With the same idea one we can show that $\forall t > s'$, $p_{jt} - p_{j(t+1)} \leq w_j \lambda_i$.

Notice that $(w_j s) \lambda_i + p_{js} = (w_j s') \lambda_i + p_{js'} = \beta_{ij} \Rightarrow p_{js} = p_{js'} + w_j \lambda_i (s' - s)$. We also have that

$$\beta_{ij} = (w_j s) \lambda_i + p_{js} \leq (w_j(s+1)) \lambda_i + p_{j(s+1)} \Rightarrow p_{j(s)} - p_{j(s+1)} \leq w_j \lambda_i$$

Therefore using Lemma 4 we get that $p_{jt} - p_{j(t+1)} = w_j \lambda_i$, $\forall s \leq t < s'$. \square

Namely, prices for any good from earlier to later slot form a non-decreasing, piecewise linear convex curve. Piecewise-linearity comes from the fact that prices are discrete and if an agent is allocated to multiple slots then the difference in consecutive prices of corresponding slots is fixed to $w_j \lambda_i$ for good j . Next we derive sufficient conditions for a feasible allocation to exist.

Definition 6 We say that an allocation \mathbf{f} is *weakly-feasible* if

$$\begin{aligned}
& \forall i, \forall j : \sum_t f_{ijt} = r_{ij} \\
& \forall j, \forall t : \sum_i f_{ijt} = 1
\end{aligned}$$

Definition 7 We say that an allocation \mathbf{f} is *feasible* if it is weakly-feasible and $\forall i, \sum_{jt} p_{jt} f_{ijt} = m_i$.

Assuming integer r_{ij} s. In what follows we assume that r_{ij} s are non-negative integers to show the main intuition without notational complexity. We will remove this assumption later. Note that for any subset of agents S if total prices of earliest $r_j(S)$ slots is less than total money $\sum_{i \in S} m_i$ then someone will be left with money even when they are allocated the earliest slot. Based on this intuition we show the following lemma.

Lemma 8 *There exists a feasible allocation \mathbf{f} iff for all subsets $S \subseteq A$ we have $\sum_{j,t \leq r_j(S)} p_{jt} \geq \sum_{i \in S} m_i$ and $\sum_{j,t \leq r_j(A)} p_{jt} = \sum_{i \in A} m_i$.*

Proof : Forward direction follows from the fact that all the initial $r_j(A)$ slots for good j are filled. For the reverse direction, suppose no weakly-feasible solution is feasible. Consider a weakly-feasible allocation \mathbf{f} that minimizes $F = \sum_i (m_i - \sum_{jt} p_{jt} f_{ijt})^2$. At \mathbf{f} there is a set of agents who are under-spending and another set who are over-spending. Denote these sets by S^+ and S^- respectively, and let S^0 be the rest. Let $\pi_j(S)$, for $S \in \{S^+, S^-, S^0\}$, be the set of slots with non-zero allocations to agents of S for good j .

For any good j if $t \in \pi_j(S^-)$, $t' \in \pi_j(S^+)$ then $t \not\prec t'$. Otherwise we can decrease F , by swapping allocation between agents of S^- and S^+ allocated to t and t' respectively. This will in fact decrease overall imbalance $\sum_i |m_i - \sum_{jt} p_{jt} f_{ijt}|$. Similarly, if $t \in \pi_j(S^0)$ and $t' \in \pi_j(S^+)$ then $t \not\prec t'$. This is because we can again do the swapping, which will keep overall imbalance the same but decreases F since F is sum of squares of surpluses. Thus we have $\forall t \in \pi_j(S^+)$ and $\forall t' \in \pi_j(A \setminus S^+)$, $t \leq t'$. Due to the fact that $r_j(S^+)$ is an integer and that \mathbf{f} is weakly-feasible, the inequality should be strict. This implies slots $\pi_j(S^+)$ are 1 through $r_j(S^+)$, and are fully bought by agents of S^+ , implying $\sum_{j,t \leq r_j(S^+)} p_{jt} < \sum_{i \in S^+} m_i$, a contradiction. \square

From Section 2 recall that market equilibrium has to satisfy optimal bundle and market clearing conditions. At prices \mathbf{p} , allocation \mathbf{f} form optimal bundle if \mathbf{f}_i is a solution of LP (1) for each agent i , and market clearing requires (15) to be satisfied. Using the fact that KKT conditions together with feasibility are sufficient for an optimal solution of an LP, next we derive sufficient conditions for λ_i s and p_{jt} s to constitute market equilibrium using the above analysis.

The basic idea is that prices constitute piecewise-linear convex, non-increasing curve, where slopes of the pieces are $-w_j \lambda_i$ s. Furthermore, to ensure optimal bundles to the agents, they are allocated to segments with whose slope is $-w_j$ times their λ_i .

Theorem 9 *Given λ and prices \mathbf{p} , such that $\lambda_1 \geq \dots \geq \lambda_{|A|} > 0$, let agents be grouped in to Q_1, \dots, Q_u sets by equality of λ_i s. For all $d \leq u$, define $\lambda^d = \lambda_i$, $i \in Q_d$, and $r_j^0 = 0$, $r_j^d = r_j(\cup_{d' \leq d} Q_{d'})$, $\forall j$. If prices satisfy,*

$$\begin{aligned} \forall j : & \quad p_{jt} > p_{j(t+1)}, \quad \forall t \leq r_j(A), \quad p_{j(r_j(A)+1)} = 0 \\ \forall j : & \quad p_{j(t-1)} - p_{jt} \geq p_{jt} - p_{j(t+1)}, \quad 1 < t \leq r_j(A) \\ \forall j, \forall d \leq u : & \quad p_{jt} - p_{j(t+1)} = w_j \lambda_i, \quad r_j^{(d-1)} < t \leq r_j^d \\ \forall d \leq u : & \quad \sum_{j, r_j^{(d-1)} < t \leq r_j^d} p_{jt} = \sum_{i \in Q_d} m_i \\ \forall d \leq u, \forall S \subset Q_d : & \quad \sum_{j, t \leq r_j(S)} p_{j(t+r_j^{(d-1)})} \geq \sum_{i \in S} m_i. \end{aligned}$$

then these are at equilibrium for market \mathcal{M} .

Proof : We first show that there exists a feasible allocation \mathbf{f} such that $\forall d, \forall i \in Q_d$ we have $f_{ijt} = 0$ if $t \notin [r_j^{(d-1)}, r_j^d]$. Using Lemma 8, $\sum_{j, r_j^{(d-1)} < t \leq r_j^d} p_{jt} = \sum_{i \in Q_d} m_i$ and the last assumption of the theorem we get that there exist a feasible allocation \mathbf{f}^d for the agents in Q_d to the slots $t \in (r_j^{(d-1)}, r_j^d]$ of each good j . Consider the allocation to be $\mathbf{f} = (\mathbf{f}^1, \dots, \mathbf{f}^u)$. In order to complete the proof, we need to show there exists an evaluation for the vector β such that the KKT conditions hold.

Claim 10 We have that $(w_j t) \lambda^d + p_{jt} = (w_j t') \lambda^d + p_{jt'}, \forall t, t' \in (r_j^{(d-1)}, r_j^d]$ and $(w_j t) \lambda^d + p_{jt} \leq (w_j t') \lambda^d + p_{jt'}, \forall t \in (r_j^{(d-1)}, r_j^d]$ and $\forall t' \notin (r_j^{(d-1)}, r_j^d]$.

Proof : Observe that $((w_j t) \lambda^d + p_{jt}) - ((w_j t') \lambda^d + p_{jt'}) = (w_j \lambda^d)(t - t') - (p_{jt'} - p_{jt})$. Let $t' = r_j^d$. Using the assumptions it is easy to show the following

$$\begin{aligned} p_{jt} - p_{jt'} &= w_j \lambda^d (t' - t), & r_j^{(d-1)} < t \leq r_j^d \\ p_{jt} - p_{jt'} &\geq w_j \lambda^d (t' - t), & t \leq r_j^{(d-1)} \\ p_{jt'} - p_{jt} &\leq w_j \lambda^d (t - t'), & t > r_j^d \end{aligned}$$

Therefore, the lemma follows. \square

Let $\beta_{ij} = (w_j r_j^d) \lambda^d + p_{jr_j^d}, \forall j \in Q^d$. Then we have $\beta_{ij} \leq (w_j t) \lambda_i + p_{ij}$ using Claim 10. Further, if $f_{ijt} > 0$ for $i \in Q_d$ then we have $t \in (r_j^{(d-1)}, r_j^d]$ therefore $\beta_{ij} = (w_j t) \lambda_i + p_{jt}$ using Claim 13. Hence the KKT conditions hold and the proof is completed. \square

Arbitrary rational r_{ij} s. For the case where non-zero r_{ij} s take arbitrary value ≥ 1 , we will extend the characterization of Theorem 9. The primary difference from the case when r_{ij} s are integer is that $r_j(S)$ may be fractional, and therefore we need to be careful while computing slots occupied by agents of S . For the characterization to go through we need to first prove the feasibility lemma with a slight change in the condition to allow fractionally allocated slots.

Lemma 11 Suppose first slot of each good j available only up to $l_j \leq 1$. There exists a feasible allocation \mathbf{f} such that $\forall i, \sum_{jt} p_{jt} f_{ijt} = m_i$ iff for all subsets $S \subseteq A$ we have

$$\sum_j p_{j1} l_j + \sum_j \left(\sum_{1 < t < r_j(S)} p_{jt} + p_j \lceil r_j(S) \rceil (r_j(S) - \lfloor r_j(S) \rfloor) \right) \geq \sum_{i \in S} m_i$$

and

$$\sum_j p_{j1} l_j + \sum_j \left(\sum_{t \leq r_j(A)} p_{jt} + p_j \lceil r_j(A) \rceil (r_j(A) - \lfloor r_j(A) \rfloor) \right) = \sum_{i \in A} m_i.$$

Proof : The proof idea is almost identical to proof of Lemma 8. Forward direction follows from the fact that all the initial $r_j(A)$ slots for good j are filled. For the reverse direction, suppose no weakly-feasible solution is feasible. Consider a weakly-feasible allocation \mathbf{f} that minimizes $F = \sum_i (m_i - \sum_{jt} p_{jt} f_{ijt})^2$. At \mathbf{f} there is a set of agents who are under-spending and another set who are over-spending. Denote these sets by S^+ and S^- respectively, and let S^0 be the rest. Let $\pi_j(S)$, for $S \in \{S^+, S^-, S^0\}$, be the set of slots with non-zero allocations to agents of S for good j .

For any good j if $t \in \pi_j(S^-)$, $t' \in \pi_j(S^+)$ then $t \not\leq t'$. Otherwise we can decrease F , by swapping allocation between agents of S^- and S^+ allocated to t and t' respectively. This will in fact decrease overall

imbalance $\sum_i |m_i - \sum_{jt} p_{jt} f_{ijt}|$. Similarly, if $t \in \pi_j(S^0)$ and $t' \in \pi_j(S^+)$ then $t \not\leq t'$ due to minimality of F at \mathbf{f} . Thus we have $\forall t \in \pi_j(S^+)$ and $\forall t' \in \pi_j(A \setminus S^+)$, $t \leq t'$. This slots 1 through $\lfloor r_j(S^+) \rfloor$, are fully bought by agents of S^+ with also $r_j(S^+) - \lfloor r_j(S^+) \rfloor$ fraction of slot $\lceil r_j(S^+) \rceil$. That implies

$$\sum_j p_{j1} l_j + \sum_j \left(\sum_{1 < t < r_j(S)} p_{jt} + p_j \lceil r_j(S) \rceil (r_j(S) - \lfloor r_j(S) \rfloor) \right) \geq \sum_{i \in S} m_i,$$

a contradiction. \square

We will accordingly modify the conditions in Theorem 9.

Theorem 12 Given λ and prices \mathbf{p} , such that $\lambda_1 \geq \dots \geq \lambda_{|A|} > 0$, let agents be grouped in to Q_1, \dots, Q_u sets by equality of λ_i s. For all $d \leq u$, define $\lambda^d = \lambda_i$, $i \in Q_d$, and $r_j^0 = 0$, $r_j^d = r_j(\cup_{d' \leq d} Q_{d'})$, $\forall j$. If prices satisfy,

$$\begin{aligned} \forall j : & \quad p_{jt} > p_{j(t+1)}, \quad \forall t \leq r_j(A), \quad p_{j(\lfloor r_j(A) \rfloor + 1)} = 0 \\ \forall j : & \quad p_{j(t-1)} - p_{jt} \geq p_{jt} - p_{j(t+1)}, \quad 1 < t \leq \lfloor r_j(A) \rfloor \\ \forall j, \forall d \leq u : & \quad p_{jt} - p_{j(t+1)} = w_j \lambda^d, \quad \forall \lfloor r_j^{(d-1)} \rfloor < t \leq \lceil r_j^d \rceil \\ \forall d \leq u : & \quad \sum_j (\sum_{r_j^{(d-1)} < t < r_j^d} p_{jt} + p_j \lceil r_j^{(d-1)} \rceil (\lceil r_j^{(d-1)} \rceil - r_j^{(d-1)}) + p_j \lceil r_j^d \rceil (r_j^d - \lfloor r_j^d \rfloor)) = \sum_{i \in Q_d} m_i \\ \forall d \leq u, \forall S \subset Q_d : & \quad \sum_j p_j \lceil r_j^{(d-1)} \rceil (\lceil r_j^{(d-1)} \rceil - r_j^{(d-1)}) + \sum_{j, t < r_j(S)} p_{j(t + \lceil r_j^{(d-1)} \rceil)} \\ & \quad + \sum_j p_j \lceil r_j(S) + r_j^{(d-1)} \rceil (r_j(S) + r_j^{(d-1)} - \lfloor r_j(S) + r_j^{(d-1)} \rfloor) \geq \sum_{i \in S} m_i \end{aligned}$$

then these are at equilibrium for market \mathcal{M} .

Proof : We first show that there exists a feasible allocation \mathbf{f} such that $\forall d, \forall i \in Q_d$ we have $f_{ijt} = 0$ if $t \notin [\lfloor r_j^{(d-1)} \rfloor, \lceil r_j^d \rceil]$. Using Lemma 8 and the last two assumptions of the theorem we get that there exist a feasible allocation \mathbf{f}^d for the agents in Q_d to the slots $t \in (r_j^{(d-1)}, r_j^d]$ of each good j . Consider the allocation to be $\mathbf{f} = (\mathbf{f}^1, \dots, \mathbf{f}^u)$. In order to complete the proof, we need to show there exists an evaluation for the vector β such that the KKT conditions hold.

Claim 13 We have that $(w_j t) \lambda^d + p_{jt} = (w_j t') \lambda^d + p_{jt'}$, $\forall t, t' \in (\lfloor r_j^{(d-1)} \rfloor, \lceil r_j^d \rceil]$ and $(w_j t) \lambda^d + p_{jt} \leq (w_j t') \lambda^d + p_{jt'}$, $\forall t \in (\lfloor r_j^{(d-1)} \rfloor, \lceil r_j^d \rceil]$ and $\forall t' \notin (\lfloor r_j^{(d-1)} \rfloor, \lceil r_j^d \rceil]$.

Proof : Observe that $((w_j t) \lambda^d + p_{jt}) - ((w_j t') \lambda^d + p_{jt'}) = (w_j \lambda^d)(t - t') - (p_{jt'} - p_{jt})$. Let $t' = r_j^d$. Using the assumptions it is easy to show the following

$$\begin{aligned} p_{jt} - p_{jt'} &= w_j \lambda^d (t' - t), \quad \lfloor r_j^{(d-1)} \rfloor < t \leq \lceil r_j^d \rceil \\ p_{jt} - p_{jt'} &\geq w_j \lambda^d (t' - t), \quad t \leq \lceil r_j^{(d-1)} \rceil \\ p_{jt'} - p_{jt} &\leq w_j \lambda^d (t - t'), \quad t > \lceil r_j^d \rceil \end{aligned}$$

Therefore, the claim follows. \square

Let $\beta_{ij} = (w_j r_j^d) \lambda^d + p_{jr_j^d}$, $\forall j \in Q^d$. Then we have $\beta_{ij} \leq (w_j t) \lambda_i + p_{ij}$ using Claim 10. Further, if $f_{ijt} > 0$ for $i \in Q_d$ then we have $t \in (\lfloor r_j^{(d-1)} \rfloor, \lceil r_j^d \rceil]$ therefore $\beta_{ij} = (w_j t) \lambda_i + p_{jt}$ using Claim 13. Hence the KKT conditions hold and the proof is completed. \square

Pareto optimality of equilibrium allocation follows from the fact that earliest slots are filled first (Lemma 4) and therefore total cost is always $\sum_{j, t \leq r_j(A)} w_j t + \sum_j \lceil r_j(A) \rceil (r_j(A) - \lfloor r_j(A) \rfloor)$. Thus, we get that the our market model satisfies First Welfare Theorem when $G^t = G$, $\forall t$.

4 Algorithm

In this section we will design a polynomial-time algorithm to find λ and \mathbf{p} that satisfy all the conditions of Theorem 12, and thereby get a market equilibrium. In this theorem we are partitioning agents by equality in λ_i s and assigning agents of a partition to a contiguous set of slots where prices form an arithmetic progression with difference being λ_i . Overall prices form a non-increasing piecewise-linear convex curve.

Our algorithm will construct each (segment) linear-piece of the price-curve and corresponding assignment of agents inductively starting from the last linear piece. In other words we will compute the set S of agents with least λ_i , and their corresponding λ value. This will give us prices of last $\lceil r_j(S) \rceil$ slots, as they form AP with $w_j \lambda_i$ as difference and last price being zero. Then we will remove agents in S and the last $\lceil r_j(S) \rceil$ slots allocated to them, and recurs. To avoid using $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$, and for notational simplicity, for now we assume that each r_{ij} is a non-negative integer. Later we will show that the entire algorithm extends easily for the general case with careful handling at the boundary of each linear-piece.

If S is the set of agents assigned to the last segment, then their λ and prices of corresponding slots can be computed as follows. We will denote this value by λ_S .

- Let $T_j = r_j(A) + 1$. Set $p_{jT_j} = 0$, and $\forall j, T_j - r_j(S) \leq t < T_j$, $p_{jt} = p_{j(t+1)} + w_j \lambda_S = (T_j - t) w_j \lambda_S$
- Then, $\sum_{j, t \geq T_j - r_j(S)} p_{jt} = \sum_{i \in S} m_i$ gives $\lambda_S = \frac{2 \sum_{i \in S} m_i}{\sum_j w_j r_j(S) (r_j(S) + 1)}$

Clearly, the set that forms the last segment has to give the least λ , i.e., $\arg \min_{S' \subseteq A} \lambda(S')$. Trying out all possible sets will take exponential time. Instead suppose we could guess λ corresponding to the last segment, then the corresponding set should be minimizer of

$$f_{\lambda, \mathbf{T}}(S) = \sum_{i \in S} m_i - \sum_{j, t \geq T_j - r_j(S)} p_{jt}, \text{ where } p_{jt} = (T_j - t) w_j \lambda, \forall j, \forall t \quad (6)$$

And the minimum value should be zero.

Lemma 14 For any given $\lambda \geq 0$ and integers $T_j, \forall j$, function $f_{\lambda, \mathbf{T}}$ is sub-modular.

Proof : For simplicity let us denote $f_{\lambda, \mathbf{T}}$ by f . Suppose $S' \subset S$ and $i \notin S$ then we have

$$\begin{aligned} f(S' \cup \{i\}) - f(S') &= m_i - \sum_{j, T_j - r_j(S') > t \geq T_j - r_j(S' \cup \{i\})} p_{jt} \\ &\geq m_i - \sum_{j, T_j - r_j(S) > t \geq T_j - r_j(S \cup \{i\})} p_{jt} \quad * \\ &= f(S \cup \{i\}) - f(S) \end{aligned}$$

Inequality (*) holds because the prices are decreasing and $r_j(S) > r_j(S')$. \square

Finding a minimizing set of a sub-modular function can be done in polynomial time [38], and therefore we can find set $S^* \in \arg \min_{S \subseteq A} f_{\lambda, \mathbf{T}}(S)$ in polynomial time. The trick now is to guess right value of λ for which we will apply a careful binary search using the fact that minimum value is zero at right λ . To find λ and corresponding set of agents for the second last segment we need to start the last prices at $(T_j - r_j(S^*)) w_j \lambda_{S^*}$ for good j , where $S^* \in \arg \min_{S \subseteq A} f_{\lambda, \mathbf{T}}(S)$. Thus in general, to find next segment all we need to know is the remaining set of agents and price of last allocated slot for each good. To incorporate this, we modify the definition of function f as follows. Let $\mathbf{p}^l = (p_1^l, \dots, p_{|G|}^l)$ be the prices of last allocated slots in each good.

$$f_{\mathbf{p}^l, \lambda, \mathbf{T}}(S) = \sum_{i \in S} m_i - \sum_{j, T_j - r_j(S) \leq t < T_j} p_{jt}, \text{ where } p_{jt} = p_j^l + (T_j - t)w_j\lambda, \forall j, \forall t \quad (7)$$

Based on this intuition we design Algorithm 1.

Algorithm 1 Subroutine to find the least segment

- 1: LeastSeg(\mathbf{p}^l, \hat{A})
 - 2: Set $\lambda^0 = 0, \lambda^1 = \max_{i \in \hat{A}} m_i, \forall j, T_j = r_j(\hat{A}) + 1$.
 - 3: Set $S^0 \in \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^0, \mathbf{T}}(S)$ and $S^1 \in \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^1, \mathbf{T}}(S)$
 - 4: **while** $\lambda_{S^0} \neq \lambda_{S^1}$ **do**
 - 5: Set $\lambda^* = \frac{\lambda^0 + \lambda^1}{2}$ and $S^* = \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^*, \mathbf{T}}(S)$.
 - 6: **if** $f_{\mathbf{p}^l, \lambda^*, \mathbf{T}}(S^*) > 0$ **then**
 - 7: Set $\lambda^0 = \lambda^*$
 - 8: **else** Set $\lambda^1 = \lambda^*$
 - 9: **end if**
 - 10: **end while**
 - 11: Return S^0 and $\lambda = \frac{2(\sum_{i \in S^0} m_i - \sum_{j \in G} r_j(S^0)p_j^l)}{\sum_{j \in G} w_j r_j(S^0)(r_j(S^0) + 1)}$.
-

It is easy to see that the next lemma follows using [38], from the description of the algorithm and the the fact that minimum difference between λ_S for any two sets of agents is at most exponentially small.

Lemma 15 *Given a set of agents A and a price vector \mathbf{p}^l , LeastSeg(\mathbf{p}^l, A) terminates in time $\text{poly}(|A|, |G|, \mathcal{L})$, where $\mathcal{L} = \max_{i \in A, j} r_{ij} + \text{bit-length of } \max_{i \in A} m_i$. Furthermore, if A is non-empty then it returns a non-empty set.*

Next we design an inductive algorithm that computes segments starting from the last, their corresponding λ and the set of agents.

Algorithm 2 Algorithm to compute equilibrium prices and λ_i s

- 1: ComputeEq
 - 2: Set $A' = A, p_j^l = 0$ and $T_j = r_j(A) + 1, p_{jT_j} = 0 \forall j \in G$. Set $k = 1$
 - 3: **while** $A' \neq \emptyset$ **do**
 - 4: $[S^k, \lambda^k] = \text{LeastSeg}(\mathbf{p}^l, A')$
 - 5: Set $p_{jt} = p_j^l + ((T_j) - t)w_j\lambda^k, \forall j \in G, \forall T_j - r_j(S^k) \leq t < T_j$.
 - 6: Set $T_j = T_j - r_j(S^k)$ and $p_j^l = p_{jT_j}, \forall j \in G$.
 - 7: Set $\forall i \in S^k, \lambda_i = \lambda^k, A' = A' \setminus S^k$, and $k = k + 1$.
 - 8: **end while**
 - 9: Output λ and \mathbf{p} .
-

Clearly the above algorithm will terminate in $O(|A|)$ iterations, as each call of LeastSeg will decrease the size of A' by at least one. Thus using Lemma 15 we get the following lemma.

Lemma 16 *Algorithm 2 terminates in time $\text{poly}(|A|, |G|, \mathcal{L})$, where $\mathcal{L} = \sum_{ij} r_{ij} + \text{bit-length of } \max_{i \in A} m_i$.*

Next we show that Algorithm 2 computes higher and higher λ in every iteration. Suppose the algorithm goes through n iteration of the while loop. Consider the S^k and λ^k generated as output of LeastSeg in k^{th} iteration.

Lemma 17 $\lambda^k \leq \lambda^{(k+1)}, \forall k < n$.

Proof : Suppose not and $\exists k < n$ where $\lambda^k > \lambda^{(k+1)}$. Let $S = S^k \cup S^{k+1}$. To get a contradiction we show the algorithm should have chosen S instead of S^k at step k . Let $\hat{\lambda}_k$ denote the value of λ_k if the algorithm chooses S instead of S^k at step k . Let $\hat{p}_j = p_j(T_j - r_j(\cup_{i < k} S^i) + 1)$, where $T_j = r_j(A)$. Note that, reversing calculation of λ in line 8 of LeastSeg for S^k , we get:

$$\sum_{i \in S^k} m_i = \sum_j (\lambda^k \frac{(r_j(S^k))(r_j(S^k) + 1)}{2} + (r_j(S^k)p_j))$$

and

$$\begin{aligned} \sum_{i \in S^{k+1}} m_i &= \sum_j (\lambda^{k+1} \frac{(r_j(S^{k+1}))(r_j(S^{k+1}) + 1)}{2} + (r_j(S^{k+1})(r_j(S^k)\lambda^k + p_j))) \\ &< \sum_j (\lambda^k \frac{(r_j(S^{k+1}))(r_j(S^{k+1}) + 1)}{2} + (r_j(S^{k+1})(r_j(S^k)\lambda^k + p_j))) \quad (\lambda^k > \lambda^{k+1}) \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{i \in S} m_i &= \sum_{i \in S^k} m_i + \sum_{i \in S^{k+1}} m_i \\ &< \sum_j (\lambda^k \frac{(r_j(S^{k+1}))(r_j(S^{k+1}) + 1) + 2(r_j(S^{k+1})r_j(S^k))}{2} + (r_j(S^{k+1})p_j)) \\ &\quad + \sum_j (\lambda^k \frac{(r_j(S^k))(r_j(S^k) + 1)}{2} + (r_j(S^k)p_j)) \\ &= \sum_j (\lambda^k \frac{(r_j(S))(r_j(S) + 1)}{2} + (r_j(S)p_j)) \end{aligned}$$

On the other hand note that

$$\sum_{i \in S} m_i = \sum_j (\hat{\lambda}_k \frac{(r_j(S))(r_j(S) + 1)}{2} + (r_j(S)p_j)).$$

So by combining the above two inequalities we get that $\hat{\lambda}_k < \lambda_k$ which is a contradiction because then the algorithm at each step k should have chosen S instead of S^k . \square

To show that Algorithm 2 computes equilibrium prices, it is enough to show that the computed p and λ satisfy all the conditions of Theorem 9.

Lemma 18 Let p and λ be output of Algorithm 2, and let $r_j^k = r_j(\cup_{k' \geq k} S^{k'})$, then

$$\begin{aligned} \forall k \leq n, \forall i \in S^k : & \quad \lambda_i = \lambda^k \\ \forall j, \forall t \leq r_j(A) : & \quad p_{jt} > p_{j(t+1)} \text{ and } p_{jr_j(A)} = 0 \\ \forall j, \forall t \leq r_j(A) : & \quad p_{j(t-1)} - p_{jt} \geq p_{jt} - p_{j(t+1)} \\ \forall j, \forall k \leq n, \forall r_j^{(k+1)} < t \leq r_j^k : & \quad p_{jt} - p_{j(t+1)} = w_j \lambda^d \\ \forall k \leq n : & \quad \sum_{j, r_j^{(k+1)} < t \leq r_j^k} p_{jt} = \sum_{i \in S^k} m_i \\ \forall k \leq u, \forall S \subset S^k : & \quad \sum_{j, t \leq r_j(S)} p_{j(t+r_j^{(k+1)})} \geq \sum_{i \in S} m_i \end{aligned}$$

Proof : We only prove the last condition holds. All the other conditions follow easily with the algorithm description. It is easy to show the last condition is equivalent to

$$\forall k \leq n, \forall S \subset S^k, \sum_{j, t \leq r_j(S)} p_{j(r_j^k - t + 1)} \leq \sum_{i \in S} m_i$$

since we have $\sum_{j, r_j^{(k+1)} < t \leq r_j^k} p_{jt} = \sum_{i \in S^k} m_i$. So we'll show this one. Suppose it doesn't hold then $\exists S \subset A, \sum_{j, t \leq r_j(S)} p_{j(r_j^k - t + 1)} > \sum_{i \in S} m_i$. We claim the algorithm should have chose S instead of S^k in k^{th} iteration because $f_{p^l, \lambda^k}(S) = \sum_{i \in S} m_i - \sum_{j, t \leq r_j(S)} p_{j(r_j^k - t + 1)} < 0$. So we get a contradiction. \square

Note that the conditions shown in Lemma 18 are exactly those of Theorem 9, where $n = u$, and $Q^d = S^{n-d+1}$. Thus using Lemmas 16 and 18, and Theorem 9 we get the main result.

Theorem 19 *Given a market \mathcal{M} , Algorithm 2 computes λ and \mathbf{p} in polynomial-time in the description of \mathcal{M} , and \mathbf{p} is an equilibrium price vector of market \mathcal{M} .*

Algorithm 2 gives equilibrium prices. To find equilibrium allocation we can solve linear feasibility problem by plugging in values of p_{jt} s into feasibility conditions, and then solving for f_{ijts} .

Remark 20 *Even though, for the case when $G^t = G, \forall t$, actual description of the market is $O(|G| + |A| + \log(\mathbf{r}) + \log(\mathbf{m}) + \log(\mathbf{w}))$, where $\log(\mathbf{v})$ for a vector \mathbf{v} is $\sum_k \log(v_k)$, the description of the equilibrium prices and allocations still requires $\sum_{ij} r_{ij}$ space as we need to decide allocation and prices of those many slots. Thus, our algorithm is polynomial-time in this description which is unavoidable.*

If we care only about equilibrium prices, then our algorithm can be modified so that it runs in time $\text{poly}(A, G, \log(\mathbf{r}) + \log(\mathbf{m}) + \log(\mathbf{w}))$; for each segment instead of storing prices explicitly, it can store corresponding set of agents, λ , and starting and ending slot for each good. Since # iterations is bounded by $|A|$ storing this information takes $O(|G| + |A| + \log(\mathbf{r}))$ space. The only difference in the While loop of the algorithm will be to not compute prices explicitly and instead just compute ending slot and its price, i.e., in step 5 compute only $p_{jT_j}, \forall j$. Similarly in LeastSeg, we can compute $f_{\mathbf{p}^l, \lambda, \mathbf{T}}(S)$ in time $O(|S| + \log(\mathbf{r}) + \log(\mathbf{m}))$ using the property that prices of good j are arithmetic progression starting at p_j^l with difference $w_j \lambda$. Thus, overall running time will be $\text{poly}(A, G, \log(\mathbf{r}) + \log(\mathbf{m}) + \log(\mathbf{w}))$.

Arbitrary rational r_{ij} s. For the case when r_{ij} s are fractional it suffices to find λ and \mathbf{p} that satisfy conditions of Theorem 12, and we will argue that a careful modification of the above algorithm works. Basically, we need to take care of partially allocated slots at the end of each iteration.

Our algorithm does allocation backwards, starting from the last slot. We need to make sure that when it terminates, the first slot of all goods are fully-allocated. Note that if $T_j = r_j(A)$ is not integral, then slot $\lceil T_j \rceil$ will be allocated up to $a_j = T_j - \lfloor T_j \rfloor$ amount and is priced at zero. We ensure this by modifying LeastSeg to incorporate a fixed subtraction from $r_j(S)$ while computing $f_{\mathbf{p}^l, \lambda, \mathbf{T}}$. In particular, in first call to LeastSeg (to compute S^1), it will subtract a_j from $r_j(S)$.

In consecutive iterations we may also need to subtract some money from $\sum_{i \in S} m_i$. This is because, imagine the earliest allocated segment in the first iteration is partially filled for some of the goods. Then agents of S^2 have to buy the rest of these slots. If price of the earliest allocated slot of good j is p_j^l and is filled up to $1 - a_j$ by agents of S^1 , then agents of S^2 will have to spend total of $\sum_j a_j p_j^l$ money to fill these. Thus, while calculating $f_{\mathbf{p}^l, \lambda, \mathbf{T}}(S)$ during the second call to LeastSeg, we need to subtract this amount from

$\sum_{i \in S} m_i$. To incorporate this redefine function f .

$$f_{\mathbf{p}^l, \lambda, \mathbf{T}, \mathbf{a}}(S) = \sum_{i \in S} m_i - \sum_j p_j^l a_j - \sum_j (\sum_{\lceil \tau_j \rceil \leq t < T_j} p_{jt} + p_j \lfloor \tau_j \rfloor (\lceil \tau_j \rceil - \tau_j)) \quad (8)$$

where $\tau_j = T_j - (r_j(S) - a_j)$, $\forall j$, and $p_{jt} = p_j^l + (T_j - t)w_j\lambda$, $\forall j, \forall t \leq T_j$

We have to do other minor modifications to take care of the fact that r_{ij} s are non-integral. The modified algorithm and subroutine are as follows. Lemmas proving its correctness and run time analysis follow almost as is.

Algorithm 3 Algorithm and its subroutine for fractional r_{ij} s.

```

1: LeastSeg( $\mathbf{p}^l, \hat{A}, \mathbf{a}$ )
2: Set  $\lambda^0 = 0, \lambda^1 = \max_{i \in \hat{A}} m_i, \forall j, T_j = \lfloor r_j(\hat{A}) \rfloor + 1$ .
3: Set  $S^0 \in \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^0, \mathbf{T}, \mathbf{a}}(S)$  and  $S^1 \in \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^1, \mathbf{T}, \mathbf{a}}(S)$ 
4: while  $\lambda_{S^0} \neq \lambda_{S^1}$  do
5:   Set  $\lambda^* = \frac{\lambda^0 + \lambda^1}{2}$  and  $S^* = \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^*, \mathbf{T}, \mathbf{a}}(S)$ .
6:   if  $f_{\mathbf{p}^l, \lambda^*, \mathbf{T}, \mathbf{a}}(S^*) > 0$  then
7:     Set  $\lambda^0 = \lambda^*$ 
8:   else Set  $\lambda^1 = \lambda^*$ 
9:   end if
10: end while
11: Calculate  $\lambda$  such that  $f_{\mathbf{p}^l, \lambda, \mathbf{T}, \mathbf{a}}(S^0) = 0$ .
12: Return  $S^0$  and  $\lambda$ 
13:
14:
15: ComputeEq
16: Set  $A' = A, p_j^l = 0$  and  $T_j = \lfloor r_j(A) \rfloor + 1, p_{jT_j} = 0, a_j = r_j(A) - \lfloor r_j(A) \rfloor, \forall j \in G$ . Set  $k = 1$ 
17: while  $A' \neq \emptyset$  do
18:    $[S^k, \lambda^k] = \text{LeastSeg}(\mathbf{p}^l, A')$ 
19:   Set  $p_{jt} = p_j^l + ((T_j) - t)w_j\lambda^k, \forall j \in G, \forall \lfloor T_j - r_j(S^k) \rfloor \leq t < T_j$ .
20:   Set  $T_j = \lfloor T_j - r_j(S^k) \rfloor, p_j^l = p_{jT_j}, a_j = r_j(A' \setminus S^k) - T_j, \forall j \in G$ .
21:   Set  $\forall i \in S^k, \lambda_i = \lambda^k, A' = A' \setminus S^k$ , and  $k = k + 1$ .
22: end while
23: Output  $\lambda$  and  $\mathbf{p}$ .
```

Note that, as per the output of the above Algorithm 3, if slot t is shared by S^k and S^{k+1} , then $p_{jt} - p_{j(t+1)} = w_j\lambda^k$ and $p_{j(t-1)} - p_{jt} = w_j\lambda^{k+1}$. Furthermore, we can show that Lemma 17 still holds, and hence price function remains piecewise-linear convex. The rest of the conditions needed in Theorem 12 follows similar to the integral case using these two observations. Thus the next result follows:

Theorem 21 Given market \mathcal{M} with $G^t = G, \forall t$, Algorithm 3 computes its equilibrium in polynomial time.

5 Case $G^t \subset G$: Characterization and Algorithm

For the case when in slot t only $G^t \subset G$ of goods are available, we need to modify the equilibrium characterization and algorithm to make sure that if good $j \notin G^t$ then it is never allocated to any agent in slot t . To

obtain a characterization for this case, we will simply omit variables corresponding to missing goods in each slot, *i.e.*, drop p_{jt} and f_{ijt} , for all $j \in G^t$. We will again construct a solution such that dual variable γ_{ijts} are all zero. To account for the absence of some goods in some slot, we define the following set consists of slots that can fulfill a specific requirement for good j starting from the first slot. We abuse the notation and define this function with respect to a set of agents, as well as a number.

Definition 22 Define $\pi_j(S) = \{t \leq \tau \mid j \in G^t\}$, where τ is an earliest slot such that $|\pi_j(S)| = \lceil r_j(S) \rceil$. Also define $\pi_j(k) = \{t \leq \tau \mid j \in G^t\}$, where τ is an earliest slot such that $|\pi_j(S)| = \lceil k \rceil$.

Furthermore, for a set of slots T , let $\max(T)$ and $\min(T)$ represent the maximum and minimum numbered slot respectively in T . It is not difficult to show counterparts of Lemmas 1, 2, 4, and 5 similarly, with following appropriate changes. Thumb-rule is to consider good j in slot t only if $j \in G^t$.

- In Lemma 1 we have $f_{ijt} = 1, \forall j, \forall t \in \pi_j(\lfloor r_{ij} \rfloor)$ and $f_{ij \max(\pi_j(\{i\}))} = r_{ij} - \lfloor r_{ij} \rfloor$, instead.
- In Lemma 4, we have $p_{jt} \geq p_{jt'}, \forall t, t' \in \pi_j(A), t < t'; p_{jt} = 0, \forall t > \max(\pi_j(A))$ instead.
- In Lemma 5, if for good j the first and last slot allocated to agent i are $s < s'$ then, for all slots t for which $j \in G^t$, if $s < t \leq s'$ then $p_{js} - p_{jt} = (t - s)w_j \lambda_i$, if $t < s$ then $p_{jt} - p_{js} \geq (s - t)w_j \lambda_i$, and if $t > s'$ then $p_{js'} - p_{jt} \leq (t - s')w_j \lambda_i$.

Thus it follows that prices will still form a piecewise-linear convex curve, with some slots and their prices missing for a good. The only ingredient remaining to formalize the sufficiency conditions is the feasibility lemma, which can be restated as follows.

Lemma 23 *There exists a feasible allocation \mathbf{f} iff for all subsets $S \subseteq A$ we have $\sum_j (\sum_{t \in \pi_j(\lfloor r_j(S) \rfloor)} p_{jt} + p_{j \max(\pi_j(S))} (r_j(S) - \lfloor r_j(S) \rfloor)) \geq \sum_{i \in S} m_i$, and $\sum_{j, t \in \pi_j(A)} p_{jt} = \sum_{i \in A} m_i$.*

Proof of the above lemma follows similar to that of Lemma 23. Finally, the theorem defining sufficient conditions for λ and \mathbf{p} to constitute market equilibrium is as follows:

Theorem 24 *Given λ and prices \mathbf{p} , such that $\lambda_1 \geq \dots \geq \lambda_{|A|} > 0$, let agents be grouped in to Q_1, \dots, Q_u sets by equality of λ_i s. For all $d \leq u$, define $\lambda^d = \lambda_i, i \in Q_d$, and $\pi_j^d = \pi_j(\lceil \pi(r_j^{(d)}) \rceil) \setminus \pi_j(\lfloor r_j^{d-1} \rfloor), \forall j$ where $r_j^0 = 0, r_j^d = r_j(\cup_{d' \leq d} Q_{d'})$. If prices satisfy,*

$$\begin{aligned}
\forall j : & p_{jt} > p_{jt'}, \quad \forall t < t', t, t' \in \pi_j(A), \quad p_{j\tau_j} = 0, \text{ where } \tau_j = \max(\pi(\lfloor r_j(A) \rfloor) + 1) \\
\forall j : & \frac{p_{jt'} - p_{jt}}{t' - t} \geq \frac{p_{jt''} - p_{jt'}}{t'' - t'}, \quad \forall t' < t < t'', t, t', t'' \in \pi_j(A) \\
\forall j, \forall d \leq u : & \frac{p_{jt} - p_{jt'}}{t' - t} = w_j \lambda^d, \quad \forall t < t', t, t' \in \pi_j^d \\
\forall d \leq u : & \sum_j (\sum_{t \in \pi_j^d} p_{jt} - p_{j \max(\pi_j(r_j^{d-1}))} (\lceil r_j^{d-1} \rceil - r_j^{d-1}) - p_{j \max(\pi_j(r_j^d))} (r_j^d - \lfloor r_j^d \rfloor)) = \sum_{i \in Q_d} m_i \\
\forall d \leq u, \forall S \subset Q_d : & \sum_j p_{j \max(\pi_j(r_j^{d-1}))} (\lceil r_j^{d-1} \rceil - r_j^{d-1}) + \sum_{j, t \in \pi_j(\lfloor r_j^{d-1} + r_j(S) \rfloor) \setminus \pi_j(r_j(S))} p_{jt} \\
& + \sum_j p_{j \max(\pi_j(r_j(S) + r_j^{d-1}))} (r_j(S) + r_j^{d-1} - \lfloor r_j(S) + r_j^{d-1} \rfloor) \geq \sum_{i \in S} m_i
\end{aligned}$$

then these are at equilibrium for market \mathcal{M} .

Next we design an algorithm to compute λ and \mathbf{p} that satisfy all the conditions of Theorem 24. In the algorithm, again we need to make sure that we talk about only available goods in every slot. Since our algorithm does allocation in backward order, we need to consider set of slots allocated to agents from back.

Furthermore, the last slot changes in every iteration of our algorithm. Therefore, w.r.t. last allocated slots \mathbf{T} define,

Definition 25 Define $\pi_j^{\mathbf{T}}(S) = \{\tau \leq t < T_j \mid j \in G^t\}$, where τ is the latest slot such that $|\pi_j(S)| = \lceil r_j(S) \rceil$. Also define $\pi_j^{\mathbf{T}}(k) = \{\tau \leq t < T_j \mid j \in G^t\}$, where τ is the latest slot such that $|\pi_j(S)| = \lceil k \rceil$.

Redefine function f to incorporate unavailability of goods in some slots.

$$f_{\mathbf{p}^l, \lambda, \mathbf{T}, \mathbf{a}}(S) = \sum_{i \in S} m_i - \sum_j p_j^l a_j - \sum_j (\sum_{t \in \pi_j^{\mathbf{T}}(\lfloor \tau_j \rfloor)} p_{jt} + p_{j \min(\pi_j^{\mathbf{T}}(\tau_j))} (\tau_j - \lfloor \tau_j \rfloor)) \quad (9)$$

where $\tau_j = (r_j(S) - a_j)$, $\forall j$, and $p_{jt} = p_j^l + (T_j - t)w_j\lambda$, $\forall j, \forall t \leq T_j, j \in G^t$

Modified algorithm and the LeastSeg subroutine, so that good j is considered in slot t only if $j \in G^t$, are given in Algorithm 4.

Algorithm 4 Algorithm and its subroutine for the case when $G^t \neq G$.

```

1: LeastSeg( $\mathbf{p}^l, \hat{A}, \mathbf{a}$ )
2: Set  $\lambda^0 = 0, \lambda^1 = \max_{i \in \hat{A}} m_i, \forall j, T_j = \max(\pi_j(\lfloor r_j(\hat{A}) \rfloor + 1))$ .
3: Set  $S^0 \in \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^0, \mathbf{T}, \mathbf{a}}(S)$  and  $S^1 \in \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^1, \mathbf{T}, \mathbf{a}}(S)$ 
4: while  $\lambda_{S^0} \neq \lambda_{S^1}$  do
5:   Set  $\lambda^* = \frac{\lambda^0 + \lambda^1}{2}$  and  $S^* = \arg \min_{S \subset \hat{A}} f_{\mathbf{p}^l, \lambda^*, \mathbf{T}, \mathbf{a}}(S)$ .
6:   if  $f_{\mathbf{p}^l, \lambda^*, \mathbf{T}, \mathbf{a}}(S^*) > 0$  then
7:     Set  $\lambda^0 = \lambda^*$ 
8:   elseSet  $\lambda^1 = \lambda^*$ 
9:   end if
10: end while
11: Calculate  $\lambda$  such that  $f_{\mathbf{p}^l, \lambda, \mathbf{T}, \mathbf{a}}(S^0) = 0$ .
12: Return  $S^0$  and  $\lambda$ 
13:
14:
15: ComputeEq
16: Set  $A' = A, p_j^l = 0$  and  $T_j = \max(\pi_j(\lfloor r_j(A) \rfloor + 1))$ ,  $p_{jT_j} = 0, a_j = r_j(A) - \lfloor r_j(A) \rfloor, \forall j \in G$ . Set  $k = 1$ 
17: while  $A' \neq \emptyset$  do
18:    $[S^k, \lambda^k] = \text{LeastSeg}(\mathbf{p}^l, A')$ 
19:   Set  $p_{jt} = p_j^l + ((T_j) - t)w_j\lambda^k, \forall j \in G, \forall t \in \pi_j^{\mathbf{T}}(r_j(S^k))$ .
20:   Set  $\forall i \in S^k, \lambda_i = \lambda^k, A' = A' \setminus S^k$ , and  $k = k + 1$ .
21:   Set  $T_j = \min(\pi_j^{\mathbf{T}}(r_j(S^k))), p_j^l = p_{jT_j}, a_j = r_j(A') - \lfloor r_j(A') \rfloor, \forall j \in G$ .
22: end while
23: Output  $\lambda$  and  $\mathbf{p}$ .
```

Running time and correctness analysis for Algorithm 4 follows similar to the case when $G^t = G, \forall t$. And we get the following theorem.

Theorem 26 Given market \mathcal{M} , Algorithm 4 computes its equilibrium in polynomial time.

Again, Pareto optimality of equilibrium allocation follows from the fact that earliest slots are filled first (Lemma 4) and therefore total cost is always $\sum_{j,t \in \pi_j(\lfloor r_j(A) \rfloor)} w_j t + \sum_j \max(\pi_j(A))(r_j(A) - \lfloor r_j(A) \rfloor)$. Thus, we get that our market model satisfies First Welfare Theorem.

6 Continuous Case

In this section we discuss the continuous case of our model where time is not divided in slots, and changes continuously. Accordingly price and allocation are functions of time. We define function $f_{ij}(t)$ to represent allocation of good j to agent i across time, and function $p_j(t)$ represents price of good j . Next we show that this case can be handled with a careful modification to our approach for the discrete case.

6.1 Equilibrium Characterization

Since it is no more possible to capture prices and allocation in finitely many variables, the optimal bundle LP (1) does not apply, and we need to re-derive the properties characterizing equilibria. Given price functions p_j s, the optimal bundle of agent i can be calculated using the following *continuous linear program* (CLP) instead, where f_{ij} s are un-known.

$$\begin{aligned} \min : & \sum_j \int_0^{r_j(A)} w_j t f_{ij}(t) \\ s.t. : & \int_0^{r_j(A)} f_{ij}(t) \geq r_{ij}, & \forall j \in G \\ & \sum_j \int_0^{r_j(A)} p_j(t) f_{ij}(t) \leq m_i \\ & 0 \leq f_{ij}(t) \leq 1, & \forall t, \forall j \in G \end{aligned} \quad (10)$$

Using the duality theory for CLP [4] we arrive at the following characterization of optimal solution for the above program which is similar to what we have in the discrete case (see Section 3). Note that since the last constraint hold for every time t , the corresponding dual variable $\gamma'_{ij}(t)$ is a function, while for first and second constraint dual variables are β'_{ij} and λ'_i respectively.

$$\begin{aligned} \forall j \in G, \forall t : & \beta'_{ij} \leq w_j t + p_j(t) \lambda'_i + \gamma'_{ij}(t) \\ \forall j \in G, \forall t : & f_{ij}(t) > 0 \Rightarrow \beta'_{ij} = w_j t + p_j(t) \lambda'_i + \gamma'_{ij}(t) \\ \forall j \in G, \forall t : & \gamma'_{ij}(t) > 0 \Rightarrow f_{ij}(t) = 1 \\ & \sum_j \int_0^{r_j(A)} p_j(t) f_{ij}(t) < m_i \Rightarrow \lambda'_i = 0 \end{aligned} \quad (11)$$

Assuming $\forall t, \forall j, \gamma'_{ij}(t) = 0$, and $\lambda'_i > 0$. Again we will generate a solution where the agent spends all of its money and thereby we have $\lambda'_i > 0$, and $\gamma'_{ij}(t) = 0$. As done in Section 3, dividing all the relevant conditions by λ'_i and renaming give the following.

$$\begin{aligned} \forall t, \forall j \in G : & \beta_{ij} \leq (w_j \lambda_i) t + p_j(t) \\ \forall t, \forall j \in G : & f_{ij}(t) > 0 \Rightarrow \beta_{ij} = (w_j \lambda_i) t + p_j(t) \\ & \lambda_i > 0 \end{aligned} \quad (12)$$

Note that the assumptions are not necessary but we will see that in the end we will still be able to find a solution satisfying the conditions. Since the KKT conditions (12) are same as (5) of the discrete case, except for continuous variables, using similar reasoning as Lemmas 2 and 4 it follows that for each good j function $p_j(t)$ has to be decreasing, piecewise-linear and convex. Furthermore, an agent is allocated to a single segment in each good, and slope of that segment has to be $-w_j \lambda_i$. The next lemma summaries these:

Lemma 27 At equilibrium p_j is a decreasing, piecewise-linear convex function for each good j . Furthermore, if the first and last time where agent i is allocated good j are s and s' , such that $s < s'$, then $\frac{\partial p_j}{\partial t}(t) = -w_j \lambda_i$, $\forall s < t < s'$; $\frac{\partial p_j}{\partial t}(t) \leq -w_j \lambda_i$, $\forall t < s$, and $\frac{\partial p_j}{\partial t}(t) \leq -w_j \lambda_i$, $\forall t > s'$.

Computing feasible allocation $f_{ij}(t)$ s in polynomial time, given such a price functions is a bit tricky due to the continuous nature of the allocation. Next we show existence of feasible allocation through a constructive proof, which also gives a polynomial-time procedure to compute one.

Definition 28 We say that an allocation \mathbf{f} is *weakly-feasible* if

$$\forall i, \forall j : \int_0^{r_j(A)} f_{ij}(t) dt = r_{ij}.$$

Definition 29 We say that an allocation \mathbf{f} is *feasible* if it is weakly-feasible and

$$\forall i : \sum_j \left(\int_0^{r_j(A)} p_j(t) f_{ij}(t) dt \right) = m_i.$$

Following is the modified versions of Lemma 8 for the continuous case, with the difference that the proof here is constructive.

Lemma 30 A feasible allocation \mathbf{f} exists iff we have

$$\forall S \subset A : \sum_j \left(\int_0^{r_j(S)} p_j(t) dt \right) \geq \sum_{i \in S} m_i \quad \& \quad \sum_j \left(\int_0^{r_j(A)} p_j(t) dt \right) = \sum_{i \in A} m_i$$

Proof : Let's start with a weakly-feasible allocation \mathbf{f} and modify it to get a feasible allocation. For agent i let $lm_i := m_i - \sum_j \left(\int_0^{r_j(A)} f_{ij}(t) p_j(t) dt \right)$, and let s_{ij} and e_{ij} denote the earliest and latest points of time that good j is allocated to i , respectively. We say $i \prec \hat{i}$ if $s_{i\hat{j}} < e_{i\hat{j}}$ for some good j . For each agent i_0 such that $lm_{i_0} > 0$ we do the following.

- While $lm_{i_0} > 0$ do
 1. Find shortest sequence of agents i_0, i_1, \dots, i_k such that $i_0 \prec i_1 \prec \dots \prec i_k$ and $lm_{i_k} < 0$.
 2. Simultaneously swap allocation between agents i_r and i_{r+1} , $\forall r \in [0, k)$ such that all lm_{i_r} s remain unchanged except lm_{i_0} and lm_{i_k} . Note that for agents i_r and i_{r+1} we have $s_{i_{r+1}\hat{j}} < e_{i_r\hat{j}}$ for some good \hat{j} . By swapping allocation of between i_r and i_{r+1} we mean swapping the earliest allocation of i_{r+1} with latest allocation of i_r for good \hat{j} .
(Note that swapping allocation between i_r and i_{r+1} would increase lm_r because the prices are decreasing. On the other hand swapping allocation between i_{r-1} and i_r would decrease lm_r . Therefore it is possible to do the swapping simultaneously with certain rations such that lm_{i_r} s remain unchanged $\forall r \in [1, k-1]$.)
 3. The swapping must be done until we have $i_0 \prec i_1 \prec \dots \prec i_k$, $lm_{i_k} < 0$ and $lm_{i_0} > 0$.

The following claim proves that we never get stuck during the procedure.

Claim 31 Suppose $lm_{i_0} > 0$, then there exists a sequence of agents such that $i_0 \prec i_1 \prec \dots \prec i_k$ and $lm_{i_k} < 0$.

Proof : Let S denote set of all agents like i where there is a sequence of agents such that $i_0 \prec i_1 \prec \dots \prec i$. If $\exists i \in S$ such that $lm_i < 0$ then we are done. Suppose not then we have $\sum_{i \in S} lm_i > 0$. Note that the interval $[0, r_j(S)]$ of each good j has been allocated to S . Therefore, we have

$$\sum_j \left(\int_0^{r_j(S)} p_j(t) dt \right) < \sum_{i \in S} m_i$$

which is a contradiction. \square

Let E denote the set of all pairs (i, \hat{i}) such that $i \prec \hat{i}$, $\hat{i} \neq i_0$ and $lm_i \geq 0$. Let S^0 denote the set of all agents i with $lm_i = 0$. The following claim shows the procedure finishes in polynomial time.

Claim 32 After each round of the loop $|E| - |S^0|$ decreases by at least one.

Proof : Note that by swapping the allocation at each round we only decrease lm_{i_0} and increase lm_{i_k} while other lm_i s remain unchanged so $|S^0|$ would not decrease during the swapping. Also note that since we choose the shortest sequence at each round, it is easy to show that for all i_r , where $r \in [1, k-1]$, we only might increase $s_{i_r, j}$ and decrease $e_{i_r, j}$. So $|E|$ would not increase during the swapping.

On the other hand, if after the swapping we have $lm_{i_0} = 0$ or $lm_{i_k} = 0$ then we increase $|S^0|$ by at least one. Else $\exists r$ such that $i_r \prec i_{r+1}$ before swapping and $i_r \not\prec i_{r+1}$ after the swapping. Therefore, we decrease $|E|$ by at least one. So in total we decrease $|E| - |S^0|$ by at least one. \square

So the procedure finishes in polynomial times. Note that at the end we have $\forall i, lm_i \leq 0$. Also using the second assumption of lemma we get $\sum_i lm_i = 0$. Therefore, $\forall i, lm_i = 0$. So we find a feasible allocation and the proof is completed. \square

Now we have all ingredients to derive sufficiency conditions characterizing equilibria. The following theorem is analogous of Theorem 9, and proof too follows similarly using Lemmas 27 and 30.

Theorem 33 Given λ and prices p , such that $\lambda_1 \geq \dots \geq \lambda_{|A|} > 0$, let agents be grouped in to Q_1, \dots, Q_u sets by equality of λ_i s. For all $d \leq u$, define $\lambda^d = \lambda_i$, $i \in Q_d$, and $r_j^0 = 0$, $r_j^d = r_j(\cup_{d' \leq d} Q_{d'})$, $\forall j$. If prices satisfy,

$$\begin{aligned} \forall j : & \quad \frac{\partial p_j}{\partial t} < 0, \quad \forall t \leq r_j(A), \quad p_j(r_j(A)) = 0 \\ \forall j : & \quad \frac{\partial^2 p_j}{\partial t^2} \leq 0, \quad 0 < t \leq r_j(A) \\ \forall j, \forall d \leq u : & \quad \frac{\partial p_j}{\partial t} = w_j \lambda_i, \quad r_j^{(d-1)} < t < r_j^d \\ \forall d \leq u : & \quad \sum_j \int_{r_j^{(d-1)}}^{r_j^d} p_j(t) dt = \sum_{i \in Q_d} m_i \\ \forall d \leq u, \forall S \subset Q_d : & \quad \sum_j \int_0^{r_j(S)} p_j(t + r_j^{(d-1)}) dt \geq \sum_{i \in S} m_i. \end{aligned}$$

then these are at equilibrium for market \mathcal{M} .

6.2 Algorithm

The algorithm for continuous case is almost identical to the discrete case. There are only some slight modifications needed in defining function f_λ and its variants to take into consideration continuously changing prices.

$$\begin{aligned} f_{\mathbf{p}^l, \lambda, \mathbf{T}}(S) &= \sum_{i \in S} m_i - \sum_j \int_{T_j - r_j(S)}^{T_j} p_j(t) dt, \text{ where } p_j(t) = p_j^l + (T_j - t)w_j\lambda, \forall j, \forall t \\ &= \sum_{i \in S} m_i - \sum_j (r_j(S) * p_j^l + \frac{r_j(S)^2 w_j \lambda}{2}) \end{aligned} \quad (13)$$

Note that Theorem 33 shows that the prices will be piecewise linear convex functions where each linear piece corresponds to one of the Q_i 's. So the output of the algorithm can just be the start and end point of linear pieces; recall Remark 20.

Algorithm 5 Algorithm and its subroutine for the continuous case

- 1: LeastSeg(\mathbf{p}^l, \hat{A})
 - 2: Set $\lambda^0 = 0, \lambda^1 = \max_{i \in \hat{A}} m_i, \forall j, T_j = r_j(\hat{A})$.
 - 3: Set $S^0 \in \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^0, \mathbf{T}}(S)$ and $S^1 \in \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^1, \mathbf{T}}(S)$
 - 4: **while** $\lambda_{S^0} \neq \lambda_{S^1}$ **do**
 - 5: Set $\lambda^* = \frac{\lambda^0 + \lambda^1}{2}$ and $S^* = \arg \min_{S \subset \hat{A}, S \neq \emptyset} f_{\mathbf{p}^l, \lambda^*, \mathbf{T}}(S)$.
 - 6: **if** $f_{\mathbf{p}^l, \lambda^*, \mathbf{T}}(S^*) > 0$ **then**
 - 7: Set $\lambda^0 = \lambda^*$
 - 8: **else** Set $\lambda^1 = \lambda^*$
 - 9: **end if**
 - 10: **end while**
 - 11: Return S^0 and $\lambda = 2 \frac{\sum_{i \in S^0} m_i - \sum_{j \in G} r_j(S^0) p_j^l}{\sum_{j \in G} w_j r_j(S^0)^2}$.
 - 12:
 - 13:
 - 14: ComputeEq
 - 15: Set $A' = A, p_j^l = 0$ and $T_j = r_j(A), p_{jT_j} = 0 \forall j \in G$. Set $k = 1$
 - 16: **while** $A' \neq \emptyset$ **do**
 - 17: $[S^k, \lambda^k] = \text{LeastSeg}(\mathbf{p}^l, A')$
 - 18: Set $p_j(t) = p_j^l + ((T_j) - t)w_j\lambda^k, \forall j \in G, \forall T_j - r_j(S^k) \leq t \leq T_j$.
 - 19: Set $T_j = T_j - r_j(S^k)$ and $p_j^l = p_j(T_j), \forall j \in G$.
 - 20: Set $\forall i \in S^k, \lambda_i = \lambda^k, A' = A' \setminus S^k$, and $k = k + 1$.
 - 21: **end while**
 - 22: Output λ and \mathbf{p} .
-

For the time complexity of Algorithm 5, if we keep track of only starting and ending price of a segment in step 18, then it computes price functions of all the goods in polynomial time. Furthermore, we also know the set of agents assigned to each segment. Then the exact allocation $f_{ij}(t)$ s can be computed in polynomial-time using the procedure described in the proof of Lemma 30. Algorithm 5 being analogous to Algorithm 4 for the discrete case, the correctness follows similarly. In other words, we get allocation and prices that satisfy all the conditions of Theorem 33, implying that these form an equilibrium.

7 Algorithm as a Truthful Mechanism

In this section we show that our algorithm is actually truthful, i.e., the buyers have no incentive to misreport m_i s or r_{ij} s. Note that, reporting lower m_i or higher r_{ij} are the only possible types of misreport. Fixing preferences of all buyers except buyer i , consider two runs of the algorithm, one where buyer i is truthful and another where she misreports her preferences. In particular, say buyer i either reports a lower budget m'_i , and/or a higher requirements r'_{ij} for good j .

Consider the first iteration in which the two runs differ, and let (S_1, λ_1) and (S_2, λ_2) be the segments found respectively in the truthful and non-truthful runs in this iteration. For any λ and any set S that does not contain i , $f_\lambda(S)$ remains the same between the two runs; for any set S that contains i , $f_\lambda(S)$ is strictly smaller in the non-truthful run. Hence, i does not belong to any of the segments found in earlier iterations, and S_2 necessarily contains i .⁴ Further, $\lambda_2 < \lambda_1$.

Let B be the set of agents who are not in one of the segments found prior to the current iteration. By definition B is the same for both the runs, and includes i , as argued in the previous paragraph. Let f^1 and f^2 be respectively the allocations output by the algorithm for the truthful and the non-truthful runs. We will show the existence of a weakly feasible allocation f' such that (1) For every buyer $i' \in B, i' \neq i$, his flow-time in f' is no higher than his flow-time in f^1 , and (2) For buyer i , his allocation in f' is the same as his allocation in f^2 .

This implies that i is no better off in the non-truthful run, because of the following reasoning. The total weighted flow time of all the buyers in B is minimized in f^1 ; in fact, any allocation that does not “waste” any resources minimizes the total weighted flow time. Therefore, the total weighted flow time of all the buyers in B cannot be lower in allocation f' , even when the flow-time for buyer i is calculated using only his actual requirements. Since no other buyer has a higher flow-time in f' , it is impossible for i to get a lower flow-time.⁵

It remains to show the existence of f' as claimed.

Case 1: $i' \in S_2$: In this case, $f'_{i'} = f^2_{i'}$. This satisfies the second requirement since $i \in S_2$. Since $\lambda_2 < \lambda_1$, every buyer in S_2 faces a smaller price, for every good and every slot in which she is allocated. For $i' \neq i$, given the same budget and the same requirements, this means that her flow-time in f_2 is only smaller, since her allocation must optimize for her flow-time given the prices (by being an optimal solution to LP (1)).

Case 2: $i' \notin S_2$: In this case, we first start with the allocation f^1 , in the slots $[1, r_j(B \setminus S_2)]$ for good j . Note that these slots have not been allocated at all in the previous case. Consider the total deficit after this allocation. This must be equal to the total amount of slots in $[1, r_j(B \setminus S_2)]$ that are allocated to buyers in S_2 by f^1 , because of weak feasibility of f^1 . Now re-allocate these empty slots in $[1, r_j(B \setminus S_2)]$ to make up for the deficit in the first step, and note that any such re-allocation only decreases the flow time.

Thus the next theorem follows:

Theorem 34 *Given m_i s and r_{ij} s reported by agents, the market mechanism that outputs prices and allocations, where prices are computed by Algorithm 4 for the reported preferences, is strategy-proof.*

⁴Consider the possibilities where $i \notin S_2$ and note that S_2 cannot be the minimizer in the non-truthful run given that S_1 is the minimizer in the truthful run.

⁵In fact, our proof shows that i is strictly worse off.

8 Generalization to Concave Rate Functions and Existence

We present here a general model of a scheduling market, for which we will show existence of equilibria. There are many interesting special cases of this very general model, and we expect that this model will provide a rich set of questions regarding computability of equilibria, and otherwise, in the future.

A buyer in the market has a set of *jobs*, say \mathcal{J}_i for buyer i . Each job k is defined by a continuous, non-decreasing, and concave, *rate function*, U_i^k . When given a bundle of goods $\mathbf{x}_{it}^k \in \mathbb{R}_+^G$ in slot t , a $U_i^k(\mathbf{x}_{it}^k)$ fraction of job k can be completed in that slot. Therefore the requirement for this job is $\sum_t U_i^k(\mathbf{x}_{it}^k) \geq 1$. The buyer wants to complete all his jobs at a cost no larger than his budget m_i , i.e., $\sum_k \sum_t \mathbf{x}_{it}^k \cdot \mathbf{p}_t \leq m_i$, and minimize the weighted flow time, $\sum_k w_i^k \sum_t t U_i^k(\mathbf{x}_{it}^k)$. The amount of good j available in slot t is c_{jt} . Market clears when there is no over demand of any good, and demand $<$ supply implies the price is 0. The market we consider in this paper is a special case where there is a job for every good with $r_{ij} > 0$, and the rate function of the job corresponding to good j is simply x_j/r_{ij} . Further, the weights in the flow-time are universal, and don't depend on the buyer.

In the generalized market, at given prices p_{jt} for a unit of good j in slot t , agent i 's optimization problem is as follows, where x_{itj}^k is the amount of good j allocated to job k in slot t :

$$\begin{aligned} \min : & \sum_k w_i^k (\sum_t t f_{it}^k) \\ \text{s.t.} & f_{it}^k = U_i^k(\mathbf{x}_{it}^k), & \forall k, \forall t \\ \text{OPT}_i(\mathbf{p}) & \sum_t f_{it}^k \geq 1, & \forall k \\ & \sum_{jt} p_{jt} (\sum_k x_{itj}^k) \leq m_i \\ & 0 \leq x_{itj}^k \leq c_{jt} \end{aligned} \quad (14)$$

Prices are said to be at equilibrium, if when every agent is given its optimal bundle, market clears. Formally,

$$\forall j, \forall t, \sum_{i,k} x_{itj}^k \leq c_{jt} \quad \& \quad \sum_{i,k} x_{itj}^k < c_{jt} \Rightarrow p_{jt} = 0 \quad (15)$$

Equilibrium is not guaranteed to exist even in standard Fisher and Arrow-Debreu market models [2], and checking existence can become hard [9, 40, 21]. For these models, the standard practice is to assume some mild sufficiency conditions that ensures existence [2, 32]. One of these conditions is non-satiation, i.e., agent can get more and more utility by consuming more. In our market too we need to assume the following non-satiation like condition to show existence of equilibrium. We also assume that U_i^k s are well-behaved in a standard sense.

Cond-NS: Every agent i has to consume some good j in at least c_{j1} amount to fulfill all its job requirements.

Cond-LC: $\forall i \in A, \forall k \in \mathcal{J}_i, U_i^k$ is γ -Lipschitz continuous for a $\gamma < \infty$.

We will show existence by defining a correspondence whose fixed-points give market equilibria. The correspondence is motivated by the construction of Vazirani and Yannakakis [40]. Given a market let T be the maximum slot requirement for any good; this can be calculated easily from U_i^k s. Let D be the set of all (\mathbf{p}, \mathbf{x}) such that $\forall (i, j, k, t), 0 \leq x_{itj}^k \leq c_{jt} + \epsilon$ and $\mathbf{p} \geq 0, \sum_{jt} c_{jt} p_{jt} \leq \sum_i m_i$. Consider correspondence $F : D \rightrightarrows D$, if $(\mathbf{p}', \mathbf{x}') \in F(\hat{\mathbf{p}}, \hat{\mathbf{x}})$, then

$$\mathbf{x}'_i \in OPT_i(\hat{\mathbf{p}}), \forall i \in A, \quad \& \quad \mathbf{p}' \in \arg \max_{\sum_{jt} c_{jt} p_{jt} \leq \sum_i m_i} \sum_{i,j,k,t} p_{jt} \hat{x}_{itj}^k \quad (16)$$

We consider prices of all the goods in slots beyond T to be zero. Vector $(\mathbf{x}^*, \mathbf{p}^*)$ is said to be a fixed point of F if $(\mathbf{x}^*, \mathbf{p}^*) \in F(\mathbf{x}^*, \mathbf{p}^*)$.

Lemma 35 *If $(\mathbf{x}^*, \mathbf{p}^*)$ is a fixed-point of F , then \mathbf{p}^* is at equilibrium in market \mathcal{M} . Furthermore, at this equilibrium all agents spend all their money.*

Proof : Market equilibrium requires two conditions to be satisfied: (i) every agent gets optimal bundle, i.e., \mathbf{x}_i is solution of OPT_i . (ii) market clears, i.e., (15). First is satisfied by definition of F given in (16), since $\mathbf{x}^* \in OPT_i(\mathbf{p}^*)$. For the second, we need to prove two conditions of (15).

Claim 36 $\forall(j, t), \sum_{ik} x_{itj}^{*k} \leq c_{jt}$

Proof : To the contrary suppose $\exists(j', t')$, s.t., $\sum_{ik} x_{it't'}^{*k} > c_{j't'} \Rightarrow \frac{\sum_{ik} x_{it't'}^{*k}}{c_{j't'}} > 1$. Now, let $q_{jt} = c_{jt} p_{jt}$ then

$$\begin{aligned} \arg \max_{\sum_{jt} c_{jt} p_{jt} \leq \sum_i m_i} \sum_{i,j,k,t} p_{jt} x_{itj}^{*k} &= \arg \max \sum_{jt} q_{jt} \leq \sum_i m_i \sum_{jt} q_{jt} \frac{\sum_{ik} x_{itj}^{*k}}{c_{jt}} \\ &\Rightarrow \max \sum_{jt} q_{jt} \leq \sum_i m_i \sum_{jt} q_{jt} \frac{\sum_{ik} x_{itj}^{*k}}{c_{jt}} > \sum_i m_i \\ (\because \text{setting } q_{j't'} &= \sum_i m_i \text{ and all others to zero gives the inequality}) \\ &\Rightarrow \sum_{jt} c_{jt} p_{jt}^* \frac{\sum_{ik} x_{itj}^{*k}}{c_{jt}} > \sum_i m_i \quad (\because \mathbf{p}^* \text{ is a maximizer}) \\ &\Rightarrow \sum_i (\sum_{jt} p_{jt}^* (\sum x_{itj}^{*k})) > \sum_i m_i \quad (\because \mathbf{p}^* \text{ is a maximizer}) \end{aligned}$$

Since $\mathbf{x}_i^* \in OPT_i(\mathbf{p}^*)$, we have $\sum_{jt} p_{jt}^* (\sum x_{itj}^{*k}) \leq m_i, \forall i$, a contradiction. \square

Claim 37 $\forall(j, t), \sum_{ik} x_{itj}^{*k} < c_{jt} \Rightarrow p_{jt}^* = 0$.

Proof : To the contrary suppose $\exists(j', t')$ s.t., $p_{j't'}^* > 0$ and $\sum_{ik} x_{it't'}^{*k} < c_{j't'}$. Then, using similar analysis as of Claim 36 shows that $\forall(j, t), \sum_{ik} x_{itj}^{*k} < c_{jt}$ and in turn $\sum_i (\sum_{jt} p_{jt}^* (\sum x_{itj}^{*k})) < \sum_i m_i$. Thus some agent is not spending all its money to buy an optimal bundle. Let this agent be i' . Now by the *Cond-NS* defined above there exists a good j which she can consume fully in first slot to fulfill her job requirements. However, for this good too is under sold in its first slot, $\sum_{ik} x_{i1j}^{*k} < c_{j1}$. So she must be given higher slot instead in \mathbf{x}_i^* . But then she can reduce her total flow time by buying good j in first slot, contradicting optimality of \mathbf{x}_i^* for $OPT_i(\mathbf{p}^*)$. \square

Thus it follows that $(\mathbf{p}^*, \mathbf{x}^*)$ constitute an equilibrium. \square

A correspondence from a closed convex set to itself has a fixed-point if every point evaluates to a closed convex set and its graph is closed [30]. Next we show the same for F .

Lemma 38 $\forall(\hat{\mathbf{p}}, \hat{\mathbf{x}}) \in D, F(\hat{\mathbf{p}}, \hat{\mathbf{x}})$ is convex and closed.

Proof : The result follows using the fact that in (16) \mathbf{p}' is a maximizer of a linear program, and $OPT_i(\hat{\mathbf{p}})$ is a convex program for all i . The latter is due to U_i^k 's being concave functions. \square

Lemma 39 *Graph of correspondence F is closed. In other words, if there are two sequences $(\mathbf{p}^{d'}, \mathbf{x}^{d'}) \rightarrow (\mathbf{p}^*, \mathbf{x}^*)$ and $(\mathbf{p}^d, \mathbf{x}^d) \rightarrow (\mathbf{p}^*, \mathbf{x}^*)$ such that $(\mathbf{p}^{d'}, \mathbf{x}^{d'}) \in F(\mathbf{p}^d, \mathbf{x}^d)$, $\forall d$, then $(\mathbf{p}^*, \mathbf{x}^*) \in F(\mathbf{p}^*, \mathbf{x}^*)$*

Proof : Let us divide F into F_1 and F_2 , where if $(\mathbf{p}', \mathbf{x}') = F(\mathbf{p}, \mathbf{x})$, then $\mathbf{p}' = F_1(\mathbf{x})$ and $\mathbf{x}' = F_2(\mathbf{p})$. By continuity of addition and max functions it follows that F_1 has a closed graph and therefore $\mathbf{p}^* \in F_1(\mathbf{x}^*)$.

For F_2 we need to understand how the solution of OPT_i changes with prices. We will show that it changes continuously using the fact that prices of goods in slots beyond T^{th} slot is zero. To the contrary suppose $\mathbf{x}^{d'} \notin F_2(\mathbf{p}^*)$. Instead let $\hat{\mathbf{x}} \in F_2(\mathbf{p}^*)$. Let $\text{cost}_i(\mathbf{x}_i)$ denote the flow-time cost (objective function of OPT_i) of agent i at bundle \mathbf{x}_i . Then, $\exists i, \delta = \text{cost}_i(\mathbf{x}_i^*) - \text{cost}_i(\hat{\mathbf{x}}_i) > 0$. Using this we will show a contradiction to $\mathbf{x}^{d'} \in F_2(\mathbf{p}^d)$, $\forall d$. Since cost_i is a continuous function, and since $\mathbf{x}^{d'} \rightarrow \mathbf{x}^*$, $\exists d'$ s.t. $\forall d \geq d', |\text{cost}_i(\mathbf{x}_i^{d'}) - \text{cost}_i(\mathbf{x}_i^*)| < \frac{\delta}{2}$ and there by $\text{cost}_i(\mathbf{x}_i^{d'}) - \text{cost}_i(\hat{\mathbf{x}}_i) > \frac{\delta}{2}$.

Define, $\text{price}(\mathbf{p}, \mathbf{x}) = \sum_{jt} p_{jt} (\sum_k x_{jt}^k)$. Since agent i is buying $\mathbf{x}_i^{d'}$ at prices \mathbf{p}^d even though $\hat{\mathbf{x}}_i$ is cost effective, it must be the case that $\hat{\mathbf{x}}_i$ is not affordable at these prices, i.e., $\text{price}(\mathbf{p}^d, \hat{\mathbf{x}}_i) > m_i$, $\forall d > d'$. On the other hand, $\hat{\mathbf{x}}_i$ is affordable at prices \mathbf{p}^* and $\mathbf{p}^d \rightarrow \mathbf{p}^*$, so there should be another sequence of $\hat{\mathbf{x}}_i^{d'} \rightarrow \hat{\mathbf{x}}_i$ such that $\hat{\mathbf{x}}_i^{d'}$ is feasible in $OPT_i(\mathbf{p}^d)$. This is using the fact that functions U_i^k , $\forall k \leq a_i$ are γ -Lipschitz continuous for $\gamma < \infty$, concave and non-decreasing, and that $\text{price}(\mathbf{p}, \mathbf{x})$ is also a continuous function. Thus, $\exists d'' > d'$ such that $|\text{cost}_i(\hat{\mathbf{x}}_i) - \text{cost}_i(\hat{\mathbf{x}}_i^{d''})| < \delta_2$, implying that $\text{cost}_i(\mathbf{x}_i^{d''}) > \text{cost}_i(\hat{\mathbf{x}}_i^{d''})$, a contradiction to $\mathbf{x}_i^{d''}$ being optimal for $OPT_i(\mathbf{p}^{d''})$. \square

The next theorem follows using Lemmas 35, 38, and 39, and the fact that a correspondence from closed convex set to itself with closed graph has a fixed-point [30].

Theorem 40 *If market \mathcal{M} satisfies Cond-NS and Cond-LC then it has an equilibrium. Furthermore, all the agents spend all their money at this equilibrium.*

Remark 41 *It is easy to construct examples where not every one spends all their money at all equilibria. Consider a market with single good available in unit amount in all the slots, and two agents, with 10 and 1 dollars respectively. Both wants to finish one job each with $U_i = x_1$ and $r_i = 1$. Then, $p_{11} = 2 + \delta, p_{12} = 1$, $\forall \delta \leq 8$ are equilibria, where first agent gets the first slot and second gets the second slot. Note that, except for $\delta = 8$, everywhere else agent 1 under spends.*

References

- [1] Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 1–7. ACM, 2006.
- [2] K. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22: 265–290, 1954.
- [3] Moshe Babaioff, Brendan Lucier, Noam Nisan, and Renato Paes Leme. On the efficiency of the walrasian mechanism. In *ACM Conference on Economics and Computation, EC*, pages 783–800, 2014.

- [4] Dimitra Bampou and Daniel Kuhn. Polynomial approximations for continuous linear programs. *SIAM Journal on Optimization*, 22(2):628–648, 2012.
- [5] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061 – 1103, 2011.
- [6] Eric B Budish and Judd B Kessler. Changing the course allocation mechanism at wharton. *Chicago Booth Research Paper*, (15-08), 2014.
- [7] X. Chen, D. Dai, Y. Du, and S.-H. Teng. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *FOCS*, 2009.
- [8] B. Codenotti, A. Saberi, K. Varadarajan, and Y. Ye. Leontief economies encode two-player zero-sum games. In *SODA*, 2006.
- [9] B. Codenotti, A. Saberi, K. Varadarajan, and Y. Ye. The complexity of equilibria: Hardness results for economies via a correspondence with games. *Theoretical Computer Science*, 408(2-3):188–198, 2008.
- [10] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. Market equilibrium via the excess demand function. In *ACM Symposium on the Theory of Computing*, 2005.
- [11] Richard Cole, Vasilis Gkatzelis, and Gagan Goel. Mechanism design for fair division: allocating divisible items without payments. In *ACM Conference on Electronic Commerce, EC '13*, pages 251–268, 2013. URL <http://doi.acm.org/10.1145/2482540.2482582>.
- [12] Xiaotie Deng, Christos Papadimitriou, and Shmuel Safra. On the complexity of equilibria. In *Proceedings of ACM Symposium on Theory of Computing*, 2002.
- [13] N. Devanur and R. Kannan. Market equilibria in polynomial time for fixed number of goods or agents. In *FOCS*, pages 45–53, 2008.
- [14] N. Devanur and V. V. Vazirani. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. In *Proceedings of 36th STOC*, 2004.
- [15] N. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani. Market equilibrium via a primal-dual algorithm for a convex program. *JACM*, 55(5), 2008.
- [16] Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78. ACM, 2009.
- [17] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Technical report, National Bureau of Economic Research, 2005.
- [18] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S Muthukrishnan. Online stochastic matching: Beating $1-1/e$. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 117–126. IEEE, 2009.
- [19] Jugal Garg, Ruta Mehta, Milind Sohoni, and Vijay V. Vazirani. A complementary pivot algorithm for market equilibrium under separable piecewise-linear concave utilities. In *ACM Symposium on the Theory of Computing*, 2012.

- [20] Jugal Garg, Ruta Mehta, and Vijay V. Vazirani. Dichotomies in equilibrium computation, and complementary pivot algorithms for a new class of non-separable utility functions. In *STOC*, 2014.
- [21] Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdabod. Leontief exchange markets can solve multivariate polynomial equations, yielding FIXP and ETR hardness. Submitted, 2015.
- [22] R. Garg and S. Kapoor. Auction algorithms for market equilibrium. In *Proceedings of 36th STOC*, 2004.
- [23] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 323–336, Berkeley, CA, USA, 2011. USENIX Association. URL <http://dl.acm.org/citation.cfm?id=1972457.1972490>.
- [24] Leonid Hurwicz. On informationally decentralized systems. In C. B. McGuire and Roy Radner, editors, *Decision and Organization: A Volume in Honor of Jacob Marschak*. North-Holland, Amsterdam, 1972.
- [25] Sungjin Im, Janardhan Kulkarni, and Kamesh Munagala. Competitive algorithms from competitive equilibria: Non-clairvoyant scheduling under polyhedral constraints. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 313–322. ACM, 2014. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591814. URL <http://doi.acm.org/10.1145/2591796.2591814>.
- [26] Sungjin Im, Janardhan Kulkarni, and Kamesh Munagala. Competitive flow-time algorithms for polyhedral scheduling. In *IEEE FOCS*. IEEE, 2014.
- [27] K. Jain. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):306–318, 2007.
- [28] Kamal Jain and Vijay V Vazirani. Eisenberg–gale markets: Algorithms and game-theoretic properties. *Games and Economic Behavior*, 70(1):84–106, 2010.
- [29] Carlee Joe-Wong, Soumya Sen, Tian Lan, and Mung Chiang. Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework. *Networking, IEEE/ACM Transactions on*, 21(6):1785–1798, 2013.
- [30] Shizuo Kakutani. A generalization of Brouwer’s fixed point theorem. *Duke Mathematical Journal*, 8(3):457–459, 1941.
- [31] F. P. Kelly and V. V. Vazirani. Rate control as a market equilibrium. Unpublished manuscript., 2002. URL <http://www.cc.gatech.edu/~vazirani/KV.pdf>.
- [32] Robert R. Maxfield. General equilibrium and the theory of directed graphs. *Journal of Mathematical Economics*, 27(1):23–51, 1997.
- [33] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.
- [34] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
- [35] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

- [36] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. Faircloud: Sharing the network in cloud computing. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 187–198, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1419-0. doi: 10.1145/2342356.2342396. URL <http://doi.acm.org/10.1145/2342356.2342396>.
- [37] Aldo Rustichini, Mark A Satterthwaite, and Steven R Williams. Convergence to efficiency in a simple market with incomplete information. *Econometrica: Journal of the Econometric Society*, pages 1041–1063, 1994.
- [38] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.
- [39] Hal R Varian. Position auctions. *international Journal of industrial Organization*, 25(6):1163–1178, 2007.
- [40] Vijay V. Vazirani and Mihalis Yannakakis. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of ACM*, 58(3):10:1–10:25, 2011.
- [41] Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *The journal of supercomputing*, 54(2):252–269, 2010.